College of Computing and Informatics
Computer Science Department
University of Sharjah
Fall 2024/2025

Project Title

EyeScan: Deep Learning Framework for Ocular Disease Diagnosis

by

Roudha Asem Alawadhi (U21103981)
Maryam Mohammed Imtiaz (U21104413)
Salha Salah Almaazmi (U20101680)

Supervised by

Dr. Karam Sallam
Dr. Ibrahim Abaker Hashem

December 2024

Abstract

The main problem is the alarming spread of eye diseases around the world. The challenge lies in the timely diagnosis of eye diseases, which often remain undetected until they become incurable. This not only highlights the major health risks but also underscores the wider impact of untreated eye conditions on society and the risks of surgical failure. The choice of this problem stems from its profound impact on individuals and nations. Late diagnosis can lead to irreversible vision loss, affecting the quality of life of millions. The main goal of this project is to solve these challenges using advanced Deep Learning (DL) algorithms. DL models are leading the charge in revolutionizing how we address these escalating issues. This project consists of three stages. In stage 1, a comparison is conducted between several existing and enhanced DL models to select the best model among these rivals. In stage 2, the best model will be enhanced and improved to boost its accuracy and performance. Additionally, the explainability of the enhanced model will be performed to enable understanding of why the model performs well in given cases. In stage 3, we will build a mobile app capable of scanning human eyes to detect the presence of certain eye diseases or confirm healthy eyes. The first stage of the project has been completed, in which we compared the performance of several DL models, namely, ResNet, ResNet with Transfer Learning (ResNet with TL), VGG19, InceptionV, InceptionV with Transfer Learning, and Neural Network. The results showed that ResNet with TL was the best in training accuracy (98%) and the second best in testing accuracy (96%) , while ResNet was the best in testing accuracy (97%).

Acknowledgements

Our thanks and appreciations go to Dr. Karam Sallam, and Dr. Ibrahim Abaker for their guidance, support, time, patience, and their useful discussions and suggestions during our work. we would also thank our families, our mothers, fathers, and our siblings, for their love and support, and all the means of comfort they provided me so that we could finish this project. We would also thank our friends who finished their Junior projects, and advised us how to deal with challenges we faced during the junior project, also we are thankful for the joyful moment that we spent together during these years in the university. We sincerely thank our grandparents, for their prayers for me and their encouragement for us to finish this project. we would not have been able to finish this project without their motivation and supplies, May ALLAH bless and provide them with health and wellness.

Abbreviations

TL	Transfer Learning
NN	Neural Network
F1	F1 Score
ODIR	Ocular Disease Intelligent Recognition
CNNs	Convolutional Neural Networks
EHR	Electronic Health Records
AUC	Area Under the Receiver Operating Characteristics Curve
AI	Artificial Intelligence
ML	Machine Learning
DED	Diabetic Eye Disease
DL	Deep Learning
RFMiD	Retinal Fundus Multi-disease Image Dataset
MGSCNN	Multilevel Glowworm Swarm-optimized Convolutional Neural Network
ResNet	Residual Network
VGG	Visual Geometry Group

Contents

Abstract iii

Acknowledgements v

Abbreviations vii

List of Figures xiii

List of Tables xvii

1 Introduction 2

1.1 Problem Statement 2

1.2 Motivation 3

1.3 Objectives 3

1.4 EyeScan's Approach 4

1.5 Addressing Limitations 4

1.6 Project Phases 4

1.7 Project Summary 5

1.8 Project timeline 6

2 Background and Literature Work 8

2.1 Introduction 8

2.2 Literature review 9

2.3 Summary 14

3 Methodology **16**

3.1 Introduction 16

3.2 Dataset 17

3.3 Dataset pre-processing: 18

3.3.1 Image cleaning 18

3.3.2 Normalization 18

3.3.3 Data Annotations 19

3.3.4 Splitting the dataset 19

3.3.5 Handling class imbalance 19

3.3.6 Dataset features 19

3.4 Deep Learning Models 20

3.4.1 VGG19 Model 20

3.4.2 ResNet Model 22

3.4.3 InceptionV3 Model 23

3.4.4 NN Model 24

3.5 Conclusion 26

4 Experiments and Results for Binary Classification **28**

4.1 Introduction 28

4.2 Experimental Configuration 28

4.3 Performance Measurements 28

4.4 Experimentation and Results 29

4.5 Discussion and Analysis 33

5 Multi-Class Classification in Eye Disease Detection Using Deep Learning **36**

5.1 Introduction 36

5.2 Multi-Class Classification in Deep Learning 36

5.3 Models for Multi-Class Image Classification 37

5.3.1 Improved Resnet50 37

5.4 Comparative Analysis of Multi-Class Models 39
5.5 Applications in Eye Disease Detection 40
5.6 Conclusion 40

6 Analysis **43**
6.1 Introduction 43
6.2 ResNet50 Model 43
6.3 ResNet50 Multi-Class Model 48
6.4 VGG-19 Multi-Class Model 53
6.5 Conclusion 58

7 Implementation **60**
7.1 Introduction 60
7.2 Prototype 60
7.3 System Development 62
7.4 EyeScan Web Page 70
7.5 Deep Learning model 75
7.6 conclusion 75

8 Conclusion **78**

A Source Code **79**

References **87**

List of Figures

1.1 Project Plan and Timeline [1] 6

3.1 Framework of the research : start with collecting the dataset, then preprocessing, then discussing different learning models, and finally classification of the dataset. 17

3.2 images from the dataset [2]. 18

3.3 Architecture of ResNet Model [1] 22

3.4 Architecture of InceptionV3 Model [3]. 24

3.5 Architecture of NN Model [4] 25

4.1 Displaying the results of accuracy, precision , recall and F1-score of the model with InceptionV3 30

4.2 Displaying the results of accuracy, precision , recall and F1-score of the model InceptionV3 without TL 30

4.3 Displaying the results of accuracy, precision , recall and F1-score of the model VGG19 31

4.4 Displaying the results of accuracy, precision , recall and F1-score of the model ResNet with TL 31

4.5 Displaying the results of accuracy, precision , recall and F1-score of the model ResNet without TL 32

4.6 Displaying the results of accuracy, precision , recall and F1-score of the model Neural Network 32

4.7 Displaying the results of all the models (a) accuracy , (b) Precision ,(c) Recall and (d) F1-Score. 34

4.8 Displaying the model accuracy and loss. Prediction errors are quantified by loss, whereas accurate predictions are measured by accuracy. Overfitting can be found by keeping an eye on accuracy and loss throughout training and validation. Increasing training accuracy but decreasing validation accuracy or decreasing training loss but increasing validation loss are signs of overfitting. Techniques like regularization and adjusting hyperparameters can mitigate overfitting, with consideration for alternative metrics like precision and recall. . . . 35

5.1 comparing different multi-class models 39

6.1 The confusion matrix for ResNet50 model. 45

6.2 The Accuracy-Loss graph for ResNet50 model. 46

6.3 The ROC curves graph for ResNet50 model. 47

6.4 The confusion matrix for ResNet50 multi-class model. 49

6.5 The Accuracy-Loss graph for ResNet50 multi-class model. 50

6.6 The ROC curve graph of ResNet50 multi-class model. 51

6.7 The confusion matrix for VGG19 multi-class model. 54

6.8 The Accuracy-Loss graph for VGG19 multi-class model. 56

6.9 The ROC graph for VGG19 multi-class model. 57

7.1 Prototype using Figma 61

7.2 Login page 62

7.3 Home page 63

7.4 upload image page 65

7.5 Diseases page 67

7.6 Diseases informations page 68

7.7 Settings page 69

7.8 Eye Disease Classifier home page. 70

7.9 Eye Disease Classifier settings page 71
7.10 Eye Disease Classifier Web Page(1). 72
7.11 Eye Disease Classifier Web Page(2). 73
7.12 Eye Disease Classifier Web Page(3). 74

A.1 Screenshot of the NN model 80
A.2 Displaying the implementation of the code that shows the confusion matrix and the accuracy. 81
A.3 Displaying the InceptionV3 model 82
A.4 Displaying the implementation of the code that shows the confusion matrix and the accuracy. 83
A.5 Displaying the implementation of the code that shows the confusion matrix and the accuracy. 84
A.6 Screenshot of the ResNet Multi-Class classification code that is used to build our mobile application(1) 85
A.7 Screenshot of the ResNet Multi-Class classification code that is used to build our mobile application(2) 85

List of Tables

4.1 Comparison between the models based on Accuracy, Precision, Recall, F1 Score 33

Chapter 1

Introduction

Eye diseases constitute a significant global health threat, affecting millions of individuals and potentially leading to irreversible vision impairment if not detected and treated promptly [5]. The subtlety of symptoms often results in late-stage diagnoses, underscoring the critical need for early detection to preserve vision [6]. Traditional diagnostic methods, relying on periodic eye exams, face challenges of accessibility and periodicity, contributing to delayed interventions. Therefore, EyeScan, our innovative initiative, addresses these challenges by leveraging a deep learning model capable of not only detecting early diagnoses and efficiently assessing various eye diseases but also achieving exceptionally high accuracy rates.

1.1 Problem Statement

Eye disorders represent a serious threat to the millions of people all around the world, posing a threat to their quality of life. The diagnostic environment of nowadays mostly depends on routine eye examinations performed by medical experts on a regular basis. However, there are a number of issues regarding this matter. Many people don't have easy access to expert eye care services, especially those who live in rural or impoverished areas. The problem of misdiagnosed and untreated eye problems is made worse by the geographic and economic disparities in healthcare availability. Because of the emphasis on routine screenings, ocular illnesses may worsen between visits and result in late-stage diagnosis with few and ineffective treatment options. Furthermore, Treatment and diagnosis may be delayed by the time-consuming nature

of traditional diagnostic techniques, which call for the patient to physically attend a medical facility. Not to mention, traditional eye care is expensive and resource-intensive due to the requirement for specialized equipment and skilled workers, which restricts the availability of regular and comprehensive exams.

In light of these difficulties, novel approaches that might offer quick, effective, and easily accessible eye disease screening are desperately needed. In order to meet this requirement, EyeScan developed to make it easier to identify different eye disorders early and accurately. This is done by utilizing cutting-edge deep learning techniques.

1.2 Motivation

EyeScan's motivation stems from the alarming rise of eye diseases globally, affecting over 290 million people, with 80 million suffering from preventable or treatable blindness [5]. This motivation is further fueled by the challenges in timely identification and treatment of eye diseases, emphasizing the need for early detection. Traditional diagnosis methods, known for their time-consuming nature, contribute to delays in treatment initiation, reinforcing the motivation behind EyeScan to leverage deep learning for efficient and timely eye disease diagnosis.

1.3 Objectives

The following are the objectives of the EyeScan project:

- Develop an enhanced Deep Learning Model: Create and hone an advanced deep learning model that can recognize a variety of eye conditions from retinal images.

- To utilize, collect and preprocess a comprehensive and diverse dataset for training and testing of the deep learning model. This dataset should include the Ocular Disease Intelligent Recognition (ODIR) dataset, with a focus on high-quality, labeled data.

- To evaluate the developed model using various performance metrics and implement the model using mobile and web-based framework

By achieving and adhering to these objectives, EyeScan aims to revolutionize the early detection and management of eye diseases, contributing to better eye health outcomes and enhancing the quality of life for individuals worldwide.

1.4 EyeScan's Approach

EyeScan aims to revolutionize early eye disease detection by redfirst developing a sophisticated model for accurate disease identification, that is used later in the second phase, on a platform to empower individuals to assess their eye health proactively. Through advanced algorithms, the model efficiently analyzes relevant images and data, putting control directly into the hands of users. This comprehensive approach not only results in a robust diagnostic tool but also introduces a convenient and accessible solution for enhanced eye disease detection.

1.5 Addressing Limitations

Recognizing the limitations in accurate disease detection models, EyeScan strategically focuses on overcoming these constraints. The project's emphasis on obtaining diverse and comprehensive datasets during the ongoing data collection phase ensures the model's adaptability and robustness. As part of this initiative, EyeScan has started working with the Ocular Disease Intelligent Recognition (ODIR) dataset, a structured ophthalmic database containing data from 5,000 patients. This dataset, representing real-life patient information, is acquired from different hospitals and medical centers in China, captured by various cameras, such as Canon, Zeiss, and Kowa, resulting in varied image resolutions.

1.6 Project Phases

EyeScan progresses through sequential phases, each crucial for the development of an effective eye disease detection system. The ongoing work includes:

- **Data Collection Phase**: Meticulously gathering relevant datasets, such as the Ocular Disease Intelligent Recognition (ODIR), to train the model effectively. The emphasis is on obtaining diverse and comprehensive data to ensure the robustness and adaptability of the system.

- **Data Preprocessing Phase**: Cleaning, organizing, and transforming the acquired dataset, including the ODIR dataset, to enhance its quality and compatibility with subsequent stages. This phase is critical as the model's effectiveness hinges on the quality of the data it is trained on.

- **Model Building and Training Phase**: Designing and implementing the model architecture, followed by extensive training using preprocessed data, including the ODIR dataset. Fine-tuning and optimization techniques are applied to enhance the model's precision in recognizing various eye diseases.

1.7 Project Summary

The EyeScan system is an innovative solution addressing critical aspects of eye care through advanced deep learning models and application technologies. It aims to detect and treat various eye diseases, raise awareness, and encourage early action. Ophthalmologists, patients, and healthcare organizations will gain a great benefit from this, as the model promotes eye health and advancing healthcare technologies. The ongoing work with the ODIR dataset represents a significant step forward, ensuring a real-life representation of patient information for training and testing the model efficiently. Annotations by trained human readers and diverse image resolutions captured by different cameras contribute to the dataset's richness and authenticity, making it a valuable resource for EyeScan's development.

1.8 Project timeline

Figure 1.1 shows the timeline of activities for our project. As depicted in this figure, the senior project begins with converting the deep learning models implemented in the junior project to handle the multi-class classification problem. This phase spans three weeks, from week 1 to week 3. After that, the models are trained, fine-tuned, and validated for three weeks, from week 4 to week 6. Then, comparison, analysis, and selection of the best model among those compared are conducted in phase 3, which lasts for three weeks. The development of the mobile and web applications is implemented next, lasting five weeks. Lastly, we would like to mention that we have been writing the thesis from the first week; the writing phase spans the entire period from week 1 to week 14.

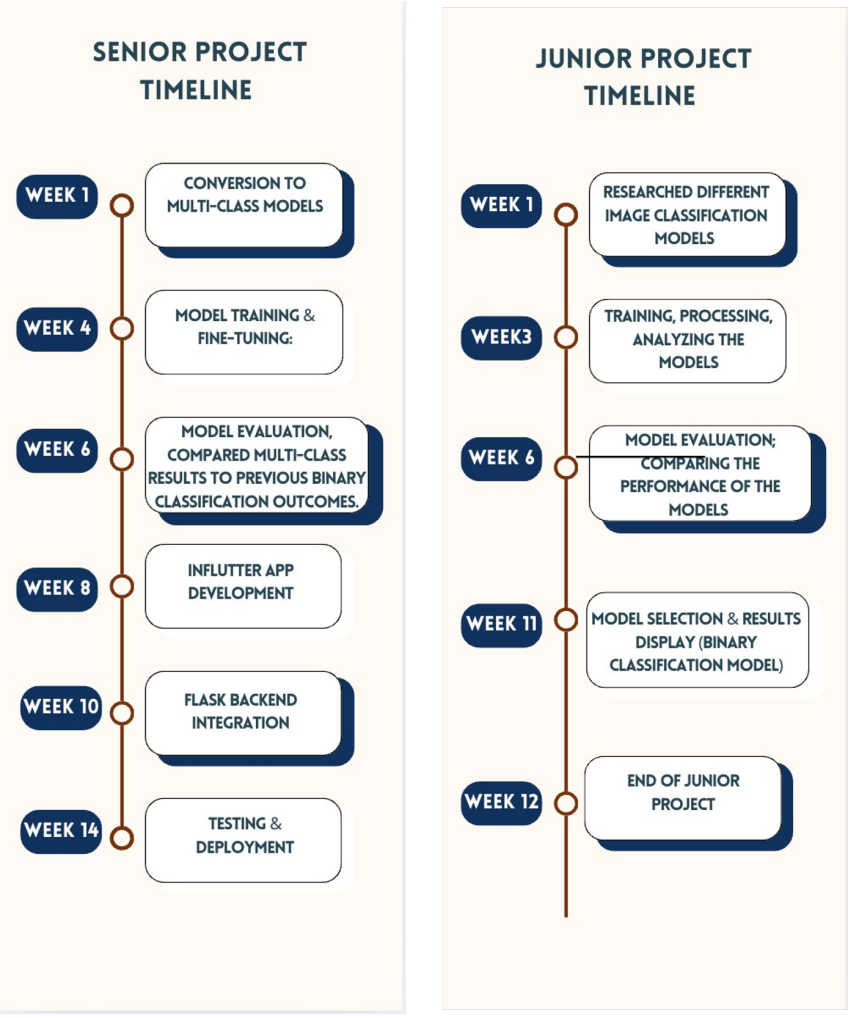


Figure 1.1: Project Plan and Timeline [1]

Chapter 2

Background and Literature Work

2.1 Introduction

This review of the literature explores a wide range of research that investigate the use of ML and DL models in the detection and classification of different eye illnesses, including as age-related macular degeneration, glaucoma, and diabetic retinopathy. This review critically evaluates the methodologies, results, and limitations of a number of studies that have made significant contributions to the field's overall objective of improving the identification and treatment of ocular diseases. Each study offers a distinct viewpoint. These research, which range from evaluations of deep learning models trained on OCT images to surveys of cutting-edge methods for multi-disease categorization, shed light on the opportunities and difficulties that arise when ophthalmology and artificial intelligence come together. The goal of this literature review is to present a thorough overview of the state of ocular disease detection with ML and DL techniques. By carefully examining methodological issues, performance measures, and real-world applications, we hope to spot new trends, point out areas that need more investigation, and emphasize how AI-driven solutions can revolutionize ophthalmic diagnostics and patient care.

2.2 Literature review

The study titled ”Using Machine Learning to Detect Different Eye Diseases from OCT Images” aims to address the critical issue of detecting retinal diseases using machine learning techniques applied to OCT images. The research utilizes the OCT-C8 dataset for training and evaluation; however, the absence of details regarding the dataset, such as the sample size and data collection process, poses a limitation to the study's transparency and reproducibility [7]. The methodology involves applying hyperparameter settings to six deep learning models, including the hyperparameter-tuned DenseNet121 model, which achieved notable performance metrics, including 97.79% accuracy, 97.69% sensitivity, and 97.79% precision for the OCT-C8 dataset. Despite the promising results, several limitations are identified in the paper. Firstly, the study focuses on a specific set of retinal diseases, potentially limiting its generalizability to the broader spectrum of eye diseases. Secondly, the comparison of the hyperparameter-tuned model's performance is confined to previous studies and lacks a comprehensive benchmark against state-of-the-art models. Lastly, the study lacks a detailed description of the OCT-C8 dataset, including potential biases and the data collection process. The omission of such information hinders the robustness of the findings and the ability to assess the dataset's representativeness.

In the paper titled ”A Survey of Deep Learning Models to Detect and Classify Eye Disorders,” the objective is to analyze deep learning and image processing-based methods for detecting and classifying eye diseases, particularly focusing on convolutional neural networks (CNNs). The study addresses gaps in existing research and identifies challenges in eye disease detection from the perspectives of image processing and machine learning [8]. The methodology involves the use of deep learning strategies, primarily CNNs, for the detection and classification of eye diseases, with a focus on improving performance through automated systems based on deep learning techniques. However, the paper lacks explicit details on the specific performance measures used to evaluate the deep learning models. The limitations of the paper are not explicitly stated, but it is crucial to recognize that the effectiveness and generalizability of the deep learning models may depend on the quality and diversity of the datasets employed,

as well as the specific characteristics of the targeted eye diseases. Additionally, it was provided that they have used a Remidio NM FOP, an AI screening tool to screen referable diabetic retinopathy and glaucoma.

The study titled ”Application of Machine Learning Predictive Models for Early Detection of Glaucoma Using Real World Data” focuses on developing a predictive analytic framework for early glaucoma detection using machine learning and logistic regression methods applied to electronic health records (EHR) data. The objective is to identify pre-glaucoma patients in advance, facilitating early intervention and preventive treatment. The study utilizes the Cerner Health Facts (r) dataset, a de-identified electronic health records database obtained from over 650 hospitals and clinics across the USA [9]. The methodology involves the application of four machine learning classification methods (XGBoost, multi-layer perceptron, random forest, logistic regression) on the entire dataset, with performance evaluation based on area under the receiver operating characteristics curve (AUC) scores. The XGBoost, multi-layer perceptron, and random forest models exhibit comparable performance, achieving an AUC score of 0.81 for predicting glaucoma one year before onset. However, logistic regression shows a lower AUC score of 0.73. Despite the comprehensive methodology and evaluation metrics, the paper lacks explicit information on its limitations. Additionally, details regarding the specific hardware and software used in the study are not provided, potentially limiting the reproducibility of the research.

In [10], the authors discusses methods for diagnosing ocular diseases using deep learning, where the researcher mainly focuses on the significance of having both accurate and gives easy to understand insights, which makes it easy and comprehensible for healthcare professionals. The researcher further investigates previous research and deep learning models for medical diagnosis to accurately diagnose several eye diseases. In simpler words, the aim is to build a computer program that can detect various types of eye diseases and explain clearly to the doctors and give the diagnosis accordingly. The paper addresses a significant issue mainly in the field of ophthalmology where giving accurate and interpretable treatments is very crucial to patient healthcare. The paper uses a dataset which comprises several ocular images including

glaucoma, diabetic retinopathy, and macular degeneration, the images for the dataset were collected in the SunYat-sen University which is a leading eye care hospital in China. 1513 images were collected and classified into 5 classes. The performance of the dataset was based on metrics like accuracy, box-plots AUC-ROC curves, which were compared to existing methods. The diagnostic had 4 stages where the first stage identifies the disease with an accuracy of 93%. The second stage locates the eye parts under two scenarios: 82% in the natural light and 90% in specific light. The third stage investigates classifying in different conditions with an accuracy of 79-98%. The labeling of the fundus images was done but the experiments wasn't completed because the output suitable for fundus images is suitable for semantic segmentation (in this process each part is label bases depending on the category which helps in understanding the structure of the eye and its features) and annotation are generated can be used effectively.

The paper titled "Automatic Detection of Diabetic Eye Disease (DED) Through Deep Learning Using Fundus Images: A Survey" gives us an overview of how deep learning is used to detect diabetic eye disease from fundus images. The researcher explores different methods, challenges, and covers preprocessing methods and datasets used for diabetic eye disease detection [11]. The main idea of the paper is the utilization of deep learning algorithms to automate the detection of the diabetic eye disease from fundus images. The author further states that through these algorithms DED can be detected at an early stage thereby reducing the risk of vision loss and any other complication associated with it (using the CNN and DBN models). The potential of deep DL techniques to automate the detection process helps in improving healthcare outcomes in diabetic retinopathy.The dataset used was a EyePACS dataset which contains a large set of fundus images, and the other dataset was a Kaggle dataset which had a retinopathy dataset, which comprises of fundus images along with corresponding labels.one limitation was the lack of diversity, limited access to high quality and diverse fundus image dataset. Deep learning models often lack interpretability, making it challenging for clinicians to understand the decision-making process behind the automated diagnosis. This lack of transparency may hinder the adoption of these models in clinical settings. Obtaining large, balanced datasets can be difficult (large annotations) Tuning parameters can be complex, and interpreting deep learn-

ing models may be challenging. Handling sensitive medical data, such as fundus images, raises concerns about privacy and security. The performance varies when using DL techniques, we look at the factors like architecture of the model, preprocessing techniques used and quality and size of the dataset used when training. Image augmentation is utilized to address imbalances in datasets, where images are mirrored, rotated, resized, and cropped to prevent overfitting and improve classification outcomes.

Another study done by John Davis Akkara and Anju Kuriakose, in his research paper "Role of artificial intelligence and machine learning in ophthalmology" [12]. The objective of the study is to explore the role of artificial intelligence (AI) and Machine Learning (ML) in ophthalmology, focusing on applications such ad diabetic retinopathy, glaucoma, age-related macular degeneration, and retinopathy of prematurity, among others. The study didn't mention a specific dataset, instead it focuses on the role of AI and ML in ophthalmology. The performance measurements of the study provide impressive results, with accuracy: 95.00%, Sensitivity: 92.5%, Specificity: 92.5%. The study had some limitations, such as data quality, the performance of AI algorithms heavily relies on the quality and quantity of data available for training, interpretability The "black box" nature of some AI algorithms, where the reasoning behind their decisions is not easily interpretable, can pose challenges in understanding how the AI arrives at its conclusions. The study mentioned that cameras such as RetCam were used, and software such as CNN and GAN.

The research paper "Deep-Ocular: Improved Transfer Learning Architecture Using Self-Attention and Dense Layers for Recognition of Ocular Diseases" done by Qaisar Abbas, Mohammed Albathan, Abdulrahman Altameem, Rami S. Almakki, and Asad Hussain, the objectives of the study include developing an automated system for identifying eye diseases effectively, integrating deep learning and ensemble methods for robust diagnostics, addressing limitations such as generalization, data availability, overfitting, limited applicability, and computational resource demands, and providing a valuable tool for healthcare professionals in diagnosing and screening eye-related diseases accurately and early [13]. The study aims to contribute to advancements in ocular disease detection and management through the proposed deep-ocular

system. The dataset used in this study consists of retinal fundus images from the Ocular Disease Intelligent Recognition (ODIR) and retinal fundus multi-disease image dataset (RFMiD). The ODIR dataset includes images of eye diseases like glaucoma, diabetic retinopathy, cataracts, and normal eyes, while the RFMiD dataset focuses on diabetic retinopathy, glaucoma, age-related macular degeneration, and other pathologies. The datasets were scaled to specific dimensions for analysis. In addition to that the performance measurements for this study can be summarized in accuracy: accuracy 90%. The limitations of the research are Limited Generalization: Performance may degrade when applied to datasets from different sources or populations due to differences in imaging quality. Data Availability: The system's performance heavily relies on the availability and diversity of high-quality labeled retinal fundus images for training, and limited data could hinder its effectiveness.Computational Resources: Training and utilizing deep learning models can demand significant computational resources, potentially limiting accessibility in resource-constrained settings [13].

Lastly, in the paper "Automatic multi-disease classification on retinal images using multilevel glowworm neural network" done by Rupali Chavan and Dnyandeo Pete. The main goal of the study is to propose a multi-disease classification method based on a multilevel glowworm swarm-optimized convolutional neural network (MGSCNN) for the classification of multiple eye diseases using retinal images [14]. The study aims to address the limitations of existing models that focus on only one eye disease and to achieve an accuracy of 95.09% through preprocessing, classification, and optimization stages. By focusing on multi-disease classification, the research aims to improve early detection of various eye diseases to prevent blindness effectively. Second, the dataset that is used in the study consists of fundus images annotations with multiple diseases. It includes 10,000 images collected from both left and right eyes of patients in 5000 clinical trials. Third, in the study, the proposed model achieved the following performance results based on the mentioned metrics: Accuracy: 95.09% Sensitivity: 96.59% Specificity: 93.59% Precision: 93.53% Recall: 96.67% F1 Score: 95.07%. Despite that there has been some limitations of the proposed model. For instance. The study has limitations related to the size, and diversity of the dataset used for training and validation. Furthermore,

Deep learning models, including convolutional neural networks, are often considered as black-box models, making it challenging to interpret how the model arrives at its decisions. Not to mention, scalability; the scalability of a proposed model to handle a larger volume of data. Moreover, the study used python to implement the proposed model in the study, as it is a popular choice for deep learning tasks due to its extensive libraries and frameworks such as TensorFlow, Keras, and PyTorch that support neural network development.

2.3 Summary

Collectively, these literature reviews contribute valuable insights into the evolving landscape of ocular disease detection. The convergence on common objectives, coupled with distinct methodological approaches, allows for a holistic understanding of the advancements and challenges in leveraging machine learning and deep learning for improved ophthalmic diagnostics. Addressing the identified gaps and limitations will be crucial for enhancing the reliability and applicability of these innovative technologies in the field of ophthalmology.

Chapter 3

Methodology

3.1 Introduction

Our project plans to create a model that helps ophthalmologists diagnose and detect diseases based on the signs and symptoms shown by the patient. Additionally, the model will provide efficient and accurate classification of retinal images for various diseases such as glaucoma, diabetic retinopathy, age-related muscular degeneration, and other ophthalmic conditions.

Figure 3.1 illustrates the framework of the research project which inlude start with collecting the dataset, then preprocessing, then discussing different learning models, and finally classification of the dataset.

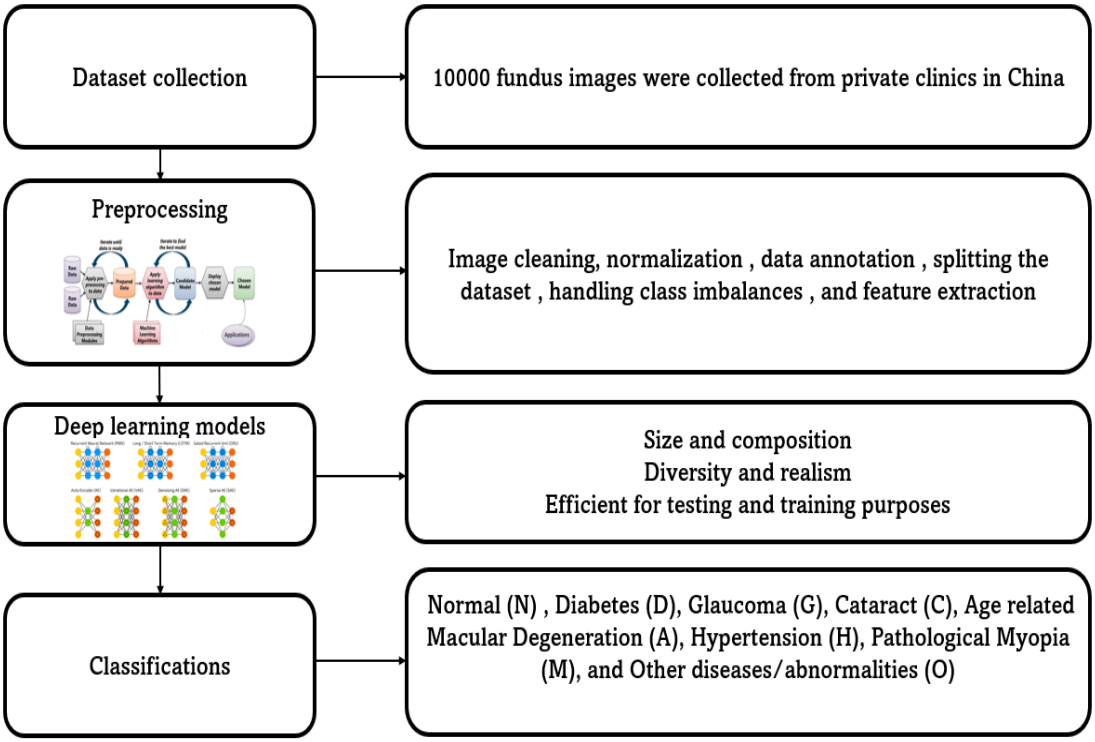


Figure 3.1: Framework of the research : start with collecting the dataset, then preprocessing, then discussing different learning models, and finally classification of the dataset.

3.2 Dataset

To understand the connection between eye disease detection and computer sciences, we read a lot of study papers and investigated many datasets. Then, we decided to choose the best dataset that has minimal limitations. We found a dataset used in deep learning ocular disease recognition research and decided to work with it. The dataset used in our research is a collection of 10,000 photos from the left and right eyes of 5,000 clinical patients making up the complete fundus image. These images show irregularities in different fundus locations caused by various illnesses. They are taken from a private clinical fundus database spanning 487 hospitals in 26 provinces in China. The dataset underwent a stringent screening procedure to ensure its high quality, and complete disease categories were consolidated for precise labeling. It aims to fill the void in ophthalmology's benchmark datasets, providing useful information for the automated categorization of clinical fundus pictures [2].

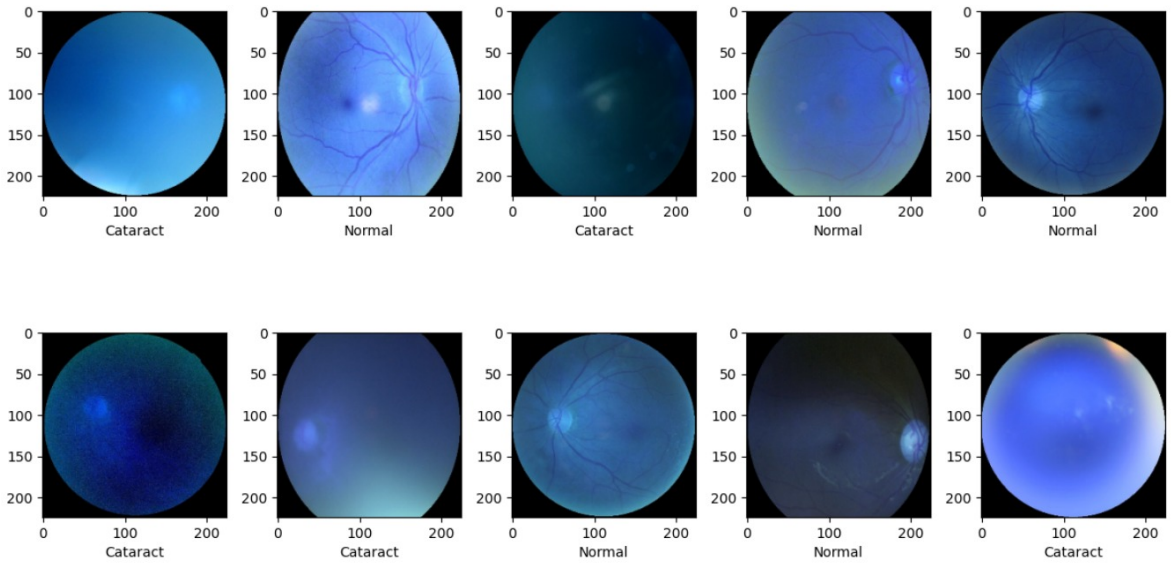


Figure 3.2: images from the dataset [2].

3.3 Dataset pre-processing:

Dataset preprocessing refers to steps and strategies applied to raw data before it is used for analysis, or model training and testing. In our dataset, the raw medical images suffer from noise, artifacts, variations in illuminations, and other inconsistencies that will affect the model's efficiency and hinder its performance. For that reason, dataset preprocessing is essential for cleaning and enhancing these images to improve their suitability for analysis [2].

3.3.1 Image cleaning

This is the initial phase of preprocessing that is mainly responsible for a meticulous cleaning process targeted at filtering out redundant and substandard images from the private clinical fundus database. By the end of this stage, the dataset will hold only high-fidelity fundus images, thereby enhancing its overall quality and reliability.

3.3.2 Normalization

In the second phase, the fundus images underwent a normalization process aimed at standardizing pixel values and intensities across the dataset. This normalization process, vital for ensuring consistency in the input data, played a primary role in optimizing the model's training process by facilitating more effective feature extraction and classification.

3.3.3 Data Annotations

This phase is pivotal in dataset preprocessing, which involves the meticulous annotations of the dataset by a team of skilled annotators adhering to stringent standards and protocols. Any inconsistencies among annotators were meticulously addressed by an arbitration team, ensuring the achievement of precise and uniform annotations across the entire dataset.

3.3.4 Splitting the dataset

Following the annotations step, the dataset was carefully divided into separate training and testing sets, maintaining an 8:2 split. The training set was used to train deep neural networks, while the off-site test set served as a validation set for model selection. Additionally, the on-site test set was carefully used to evaluate the models' ability to generalize, ensuring a comprehensive examination and validation of model performance.

3.3.5 Handling class imbalance

A notable issue that arose during preprocessing was the discovery of class imbalance in the dataset, where some classifications of eye diseases had significantly fewer instances compared to others. Interestingly, less than one-tenth of the fundus images showing normal eyes were found to depict hypertension, presenting a significant obstacle for multi-label disease detection. As a result, careful work was done to address the class imbalance, an important project aimed at supporting the development of impartial and accurate models capable of effectively detecting ocular diseases in various categories.

3.3.6 Dataset features

The main features of our dataset are size and composition. The dataset consists of 10000 fundus images across various fundus areas. Moreover, the diversity and realism in our dataset reflect the real-world clinical scenario, which will help for training and testing purposes to classify various abnormalities commonly encountered in clinical practice.

In conclusion, these meticulously executed preprocessing methods were essential for organizing the fundus images dataset for subsequent model training, testing, and evaluation. The

dataset was attentively optimized through assistance with cleaning, normalizing, annotating, splitting, and addressing class imbalances. This laid a solid foundation for the efficient training and testing of deep learning models designed for the detection and classification of eye diseases.

3.4 Deep Learning Models

To understand how deep learning models work to detect eye diseases, we examined different models including VGG19, RESNET, InceptionV3, and NN model. Some models utilize transfer learning, while others do not. All models were run using the same dataset and parameters: training=0.8, testing=0.2, batch-size=32, random-state=42, and epoch=25. During training, an epoch constitutes one complete pass over the dataset. Additionally, early stopping conditions were removed from all models and code was added to calculate performance metrics like F1 score, precision, accuracy, and recall.

3.4.1 VGG19 Model

When it comes to the deep learning model VGG19 [15], we discuss the main feature of the model which is predicting cataracts. Perhaps the code is customized for implementing VGG19 especially for this scenario, including elements such as the VGG19 architecture, pre-trained weights, and transfer learning methods. Enhancing predictive accuracy and simplifying training procedure are the primary objectives of these features. There are some limitations that affect the performance, such as the high computational cost of training deep neural networks (such as VGG19), the critical reliance on the quality and quantity of the cataract dataset used, and the challenges involved in fine-tuning pretrained models to adjust to cataract-specific features. Convolutional Neural Networks (CNNs), which are essential for image-related tasks, in addition to the implementation of the VGG19 architecture and transfer learning methods, are probably among the key algorithms included in the code. VGG19 is characterized by a uniform architecture and simplicity as shown in Figure **??**. Its primary component is a stack of three 3x3 convolutional layers. VGG19, in particular, has 19 layers total—16 convolutional layers and 3 fully linked layers. VGG networks utilize a small receptive field (3x3 convolutional kernel) and a minimum stride (1 pixel) to draw attention to depth over width. Due to these design

choices, VGG networks can develop complex characteristics at every layer [15]. Project Timeline Overview Junior Year (6 Weeks Total) Goal: Develop and evaluate binary classification models to select the highest-performing one.

Week 1-2:

Model Selection Setup: Researched and set up binary classification models: VGG19, ResNet50, InceptionV3 (with/without transfer learning), and a basic Neural Network. Week 3-4:

Data Processing Training: Collected and preprocessed dataset. Trained each binary model and monitored performance metrics. Week 5:

Model Evaluation: Compared performance and conducted cross-validation. Week 6:

Model Selection Results Display: Chose the highest-accuracy model and documented results. Senior Year (10 Weeks Total) Goal: Extend to multi-class classification and develop a Flutter app with Flask integration.

Week 1-2:

Conversion to Multi-Class: Adapted ResNet50, Improved ResNet50, and VGG19 for multi-class classification. Week 3-5:

Training Fine-Tuning: Trained multi-class models and optimized hyperparameters. Improved ResNet50 with enhancements like data augmentation. Week 6:

Evaluation Model Selection: Evaluated performance using metrics like accuracy, precision, and F1 score. Week 7-8:

Flutter App Development: Developed the mobile and web application using Flutter. Designed the UI/UX for image upload and result display. Week 9:

Flask Backend Integration: Set up a Flask API to host the multi-class Improved ResNet50 model. Implemented endpoints for image upload and model prediction. Week 10:

Testing Deployment: Integrated Flask API with Flutter frontend. Conducted testing to ensure functionality across mobile and web platforms. Deployed the app and finalized documentation.

3.4.2 ResNet Model

Developing the deep learning model ResNet has some challenges, such as designing a model for tasks related to eyes, such as classifying images or detecting objects. The code incorporates an intricate combination of essential algorithms that are required for these sorts of tasks. Convolutional Neural Networks (CNNs), which extract hierarchical properties from ocular images, are fundamental. Residual Networks (ResNet), that employ skip connections to address the problem of disappearing gradients, are vital for improving this architecture and contributing to the training of deep neural networks. By adapting a pretrained model on a larger dataset to the complicated nature of ocular images, transfer learning boosts performance. Gradient descent optimization combined with backpropagation allows continual model parameter adjustment to reduce loss and enhance prediction accuracy. While these algorithms work well together to enhance the deep learning ResNet model, it's important to keep in mind that there may be limitations related to processing power, dataset quality, and task-specific requirements. The deep architecture of the model is capable of training neural networks with hundreds of layers without encountering issues such as vanishing gradients or performance degradation. ResNet's breakthrough development is the development of skip connections, or "shortcut connections." These connections enable the network to learn residual functions. The model successfully reduces the vanishing gradients, thereby making it easier to train deeper networks by allowing input from previous layers to bypass multiple levels and reach deeper layers directly [15]. The architecture of the ResNet model is introduced in Figure 3.3.

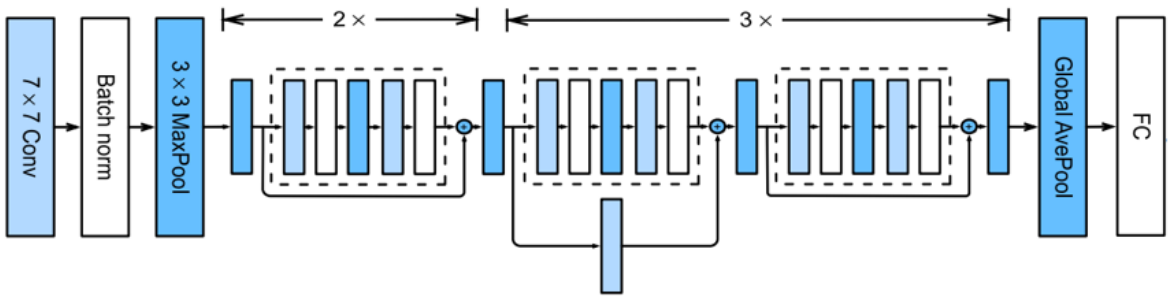


Figure 3.3: Architecture of ResNet Model [1]

3.4.3 InceptionV3 Model

In the InceptionV3 model, the main feature is predicting cataracts. Moreover, the other features are InceptionV3 architecture is adept at classifying images; it can apply transfer learning techniques and adapt models to cataract-related data (like eye cues); it can use pre-trained weights to expedite training with possibly higher accuracy; and it can employ effective training strategies to enhance model performance. In addition to that, the model involves applying cross-learning approaches to possibly adjust the model to cataract-related data, as well as utilizing the superb InceptionV3 architecture as shown in Figure 7.12, which is well-known for its proficiency in picture classification tasks. Likewise, subjects can maximize model performance by employing optimal training procedures and pre-trained weights to expedite training and increase accuracy. However, the model has limitations as the previous models, such as computational power needed to train InceptionV3-style deep neural networks, the importance of high-quality data in assessing the performance of models, and the difficulties in tailoring pretrained models to the specifics of the cataract dataset. Convolutional Neural Networks (CNNs) for hierarchical feature extraction, the special InceptionV3 architecture for effective feature capture, and the use of transfer learning to improve model performance with sparse cataract-specific data are probably among the key techniques at play. One notable feature of InceptionV3 is the use of ”Inception modules,” which are composed of many concurrent convolutional paths with different widths. To gather features at different dimensions, Inception modules utilize multiple filter sizes (e.g., 1x1, 3x3, 5x5) simultaneously, rather than depending on a single convolutional operation size. InceptionV3 also uses factorization along with additional approaches to enhance the computing economy while maintaining high precision on tasks like object identification and picture categorization. Achieving equilibrium between computing efficiency and model size is the goal of its design [3].

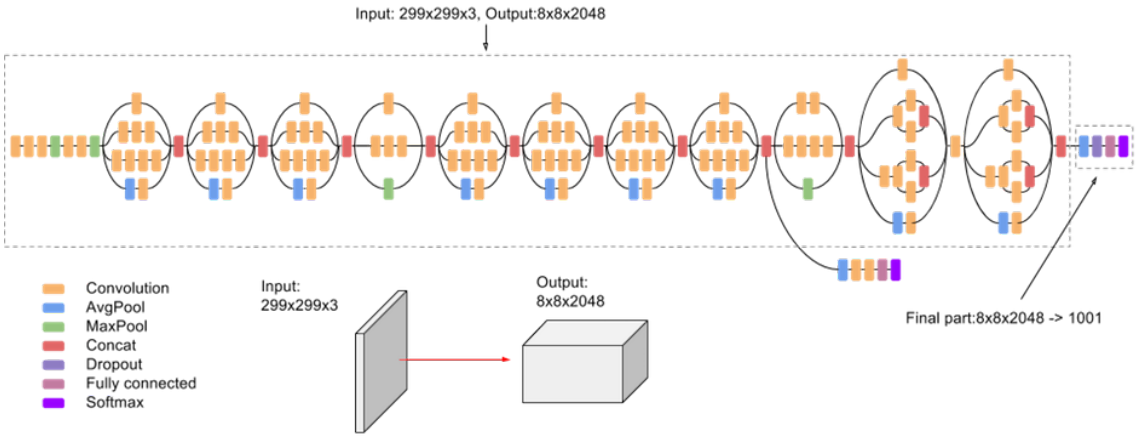


Figure 3.4: Architecture of InceptionV3 Model [3].

3.4.4 NN Model

The last model we used to test the dataset is the NN deep learning model. It uses cutting-edge deep learning algorithms to improve predicted accuracy while delving extensively into ocular-related activities. The code probably includes well-known architectures that are popular for their ability to analyze images, such as ResNet or VGG19. The application of transfer learning techniques enables the model to adjust pretrained weights for greater efficiency in ocular tasks, which is an essential aspect of this study. In addition, effective training techniques are probably used to optimize model performance and accuracy. Nevertheless, a number of difficulties might provide constraints, such as the complexity of fine-tuning pretrained models and the dependence on the quality of the dataset. Convolutional Neural Networks (CNNs) are the elementary algorithms used in this project. ResNet and VGG19 architectures are used for deep learning tasks, transfer learning is used to take advantage of prior knowledge, and optimization techniques like back propagation and gradient descent are used for model improvement. The neural network model's architecture typically consists of input, hidden, and output layers, among other layers. Weights control the connections between the neurons that make up each layer. The architecture will depend on whether it is a feed forward neural network, convolutional neural network (CNN), recurrent neural network (RNN), or one of the more advanced designs like ResNet, VGG, or Inception. Activation functions, the number of neurons in each layer, the arrangement of the layers, and other elements define the neural network model's ar-

chitecture. [16].

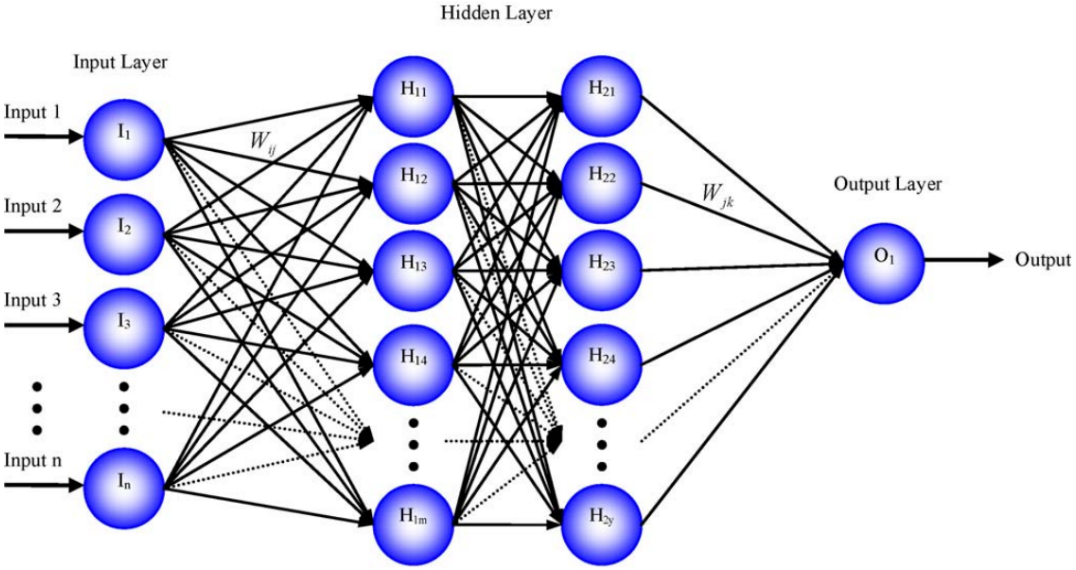


Figure 3.5: Architecture of NN Model [4]

3.5 Conclusion

In this chapter, we detailed the comprehensive methodology employed in our project to develop a model for the diagnosis and classification of eye diseases using retinal images. We began by outlining the dataset, which comprises 10,000 high-quality fundus images from a diverse pool of clinical patients. This dataset underwent rigorous preprocessing, including image cleaning, normalization, data annotation, and handling class imbalances, to ensure its suitability for model training and testing.

We then explored several deep learning models—VGG19, ResNet, InceptionV3, and a neural network (NN) model. Each model was evaluated on the same dataset using consistent parameters, allowing us to compare their performance effectively. We discussed the strengths and limitations of each architecture, emphasizing the importance of transfer learning and the unique features that each model brings to the task of ocular disease detection.

Overall, this chapter establishes a solid foundation for understanding the methodologies applied in our research, setting the stage for subsequent analysis and evaluation of model performance in future chapters. The structured approach to dataset preparation and model selection underscores our commitment to developing an efficient and accurate diagnostic tool for ophthalmologists, ultimately aiming to enhance patient outcomes in eye care.

Chapter 4

Experiments and Results for Binary Classification

4.1 Introduction

This chapter discusses the experiments that were conducted to assess our developed model when it comes to the identification and classification of eye diseases using the EyeScan dataset. We go over the tools, environments, programming languages, and technologies used, along with the processes for training and assessing each model.

4.2 Experimental Configuration

To run our experiments we have used Google Colab, a cloud-based Jupyter notebook environment that offers access to high-performance computing resources—including GPUs.GPUs are necessary for effectively training deep learning models. The Python programming language was used for model construction and evaluation, coupled with packages like PyTorch and FastAI.

4.3 Performance Measurements

Several performance indicators are used to assess a classification model's efficacy in binary classification, where the result may be categorized into one of two categories. The four typi-

cally used metrics—accuracy, precision, recall (sensitivity), and F1 score—are explained in this section along with calculations.

Accuracy : This statistic assesses how accurate the model is overall; what percentage of cases are properly predicted out of all instances, which is calculated using Equation 4.1.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{4.1}$$

Precision : Precision expresses the percentage of actual positive predictions among all the model's positive predictions. It serves as a gauge for how accurate the favorable forecasts. Equation 4.2 is utilized to calculate the value value of the precision.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives + False Positives}} \tag{4.2}$$

Recall (Sensitivity): The percentage of accurate positive predictions among all real positive examples in the dataset is measured by recall. It's a metric for how many real positives the model accurately recognized.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}} \tag{4.3}$$

F1 Score: The harmonic mean of recall and accuracy is the F1 score. It offers a harmony between recall and accuracy. Because it accounts for both false positives and false negatives, it is helpful in situations where the distribution of classes is not uniform.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

4.4 Experimentation and Results

We have employed an 80:20 train-test split for training and testing four distinct deep learning models, with the test set used for evaluation and the training set for model training. Furthermore, an on-site test set was used to evaluate the generalization capacities of the models.

The following figures illustrates the summary of the results obtained from each model:

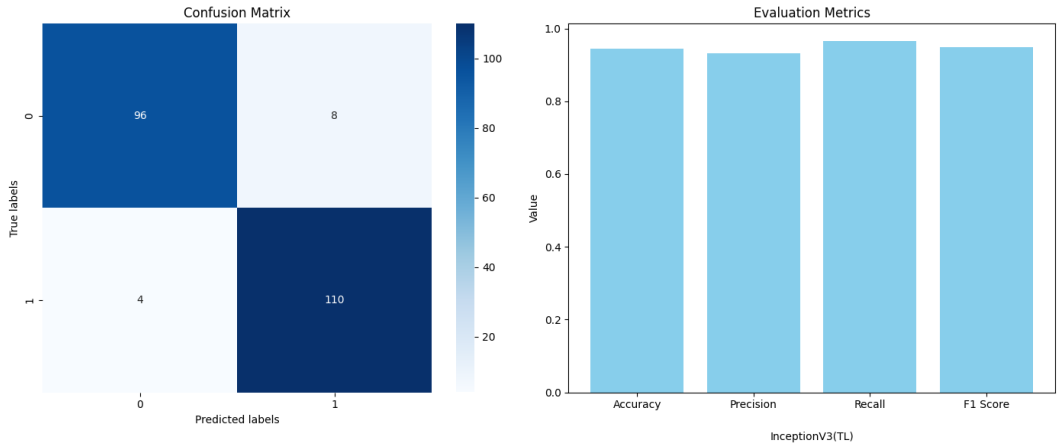


Figure 4.1: Displaying the results of accuracy, precision , recall and F1-score of the model with InceptionV3

Figure 4.1 shows the results of testing and training the InceptionV3 model with transfer learning. The value that we obtained for Accuracy was 0.94, Precision 0.93, Recall 0.96, and F1 Score 0.94.

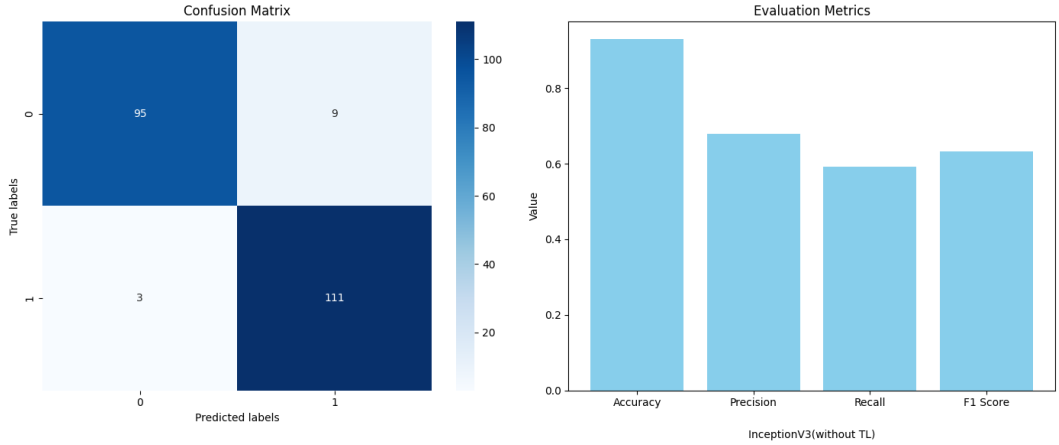


Figure 4.2: Displaying the results of accuracy, precision , recall and F1-score of the model InceptionV3 without TL

InceptionV3 without the transfer learning as shown in figure 4.2 displays the value of accuracy with 0.94, precision 0.925 recall 0.973 and f1 score 0.948.

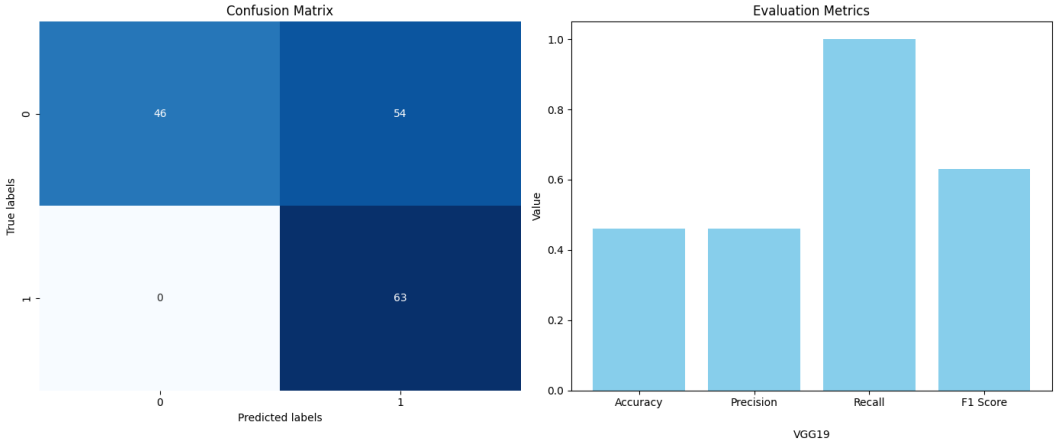


Figure 4.3: Displaying the results of accuracy, precision , recall and F1-score of the model VGG19

The values we obtained from the VGG19 shown in figure 4.3 displays low rates of accuracy with the value 0.46, precision 0.46, recall 1.00, and f1 score 0.63.

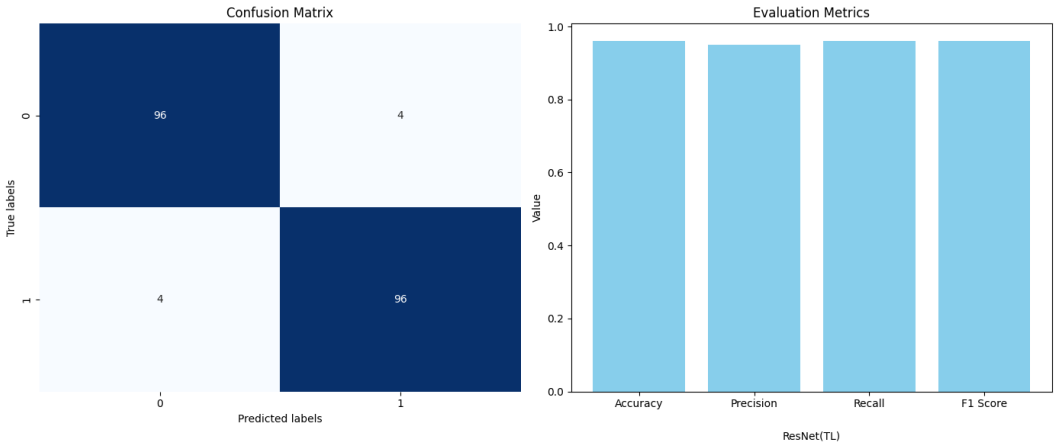


Figure 4.4: Displaying the results of accuracy, precision , recall and F1-score of the model ResNet with TL

Figure 4.4 presents the ResNet model with transfer learning. The values collected are as follows: accuracy 0.96, precision 0.95, recall 0.96, f1 score 0.96.

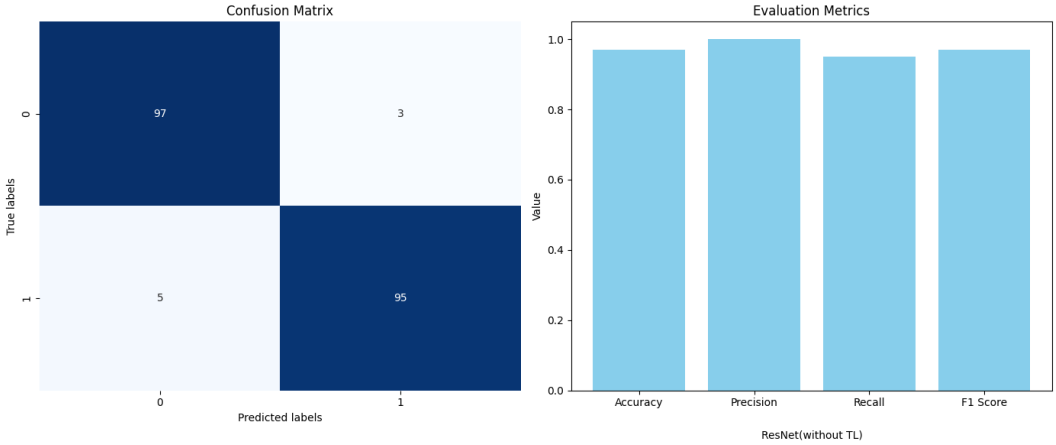


Figure 4.5: Displaying the results of accuracy, precision , recall and F1-score of the model ResNet without TL

Accuracy for the ResNet model without the transfer learning obtained is 0.97, precision is 1.00, recall is 0.95, f1 score is 0.97, as illustrated in figure 4.5.

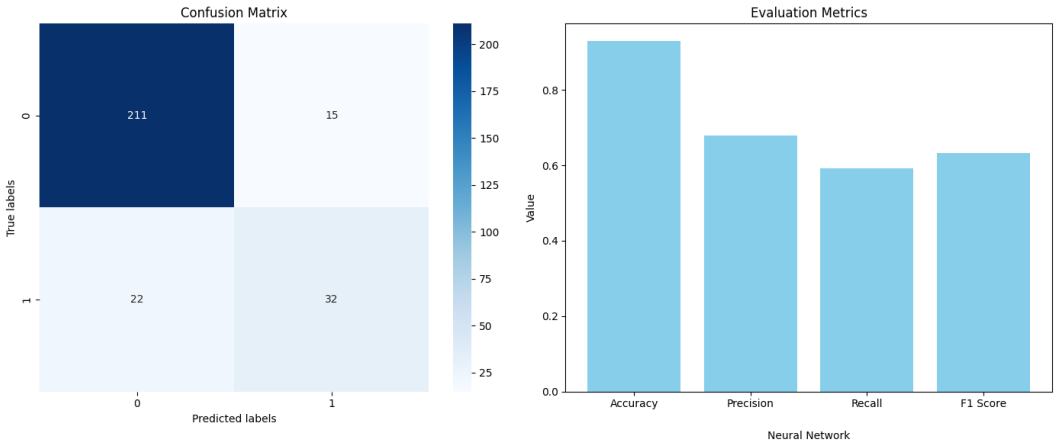


Figure 4.6: Displaying the results of accuracy, precision , recall and F1-score of the model Neural Network

As for the results of the Neural Network model displayed in figure 4.6, the value we acquired for accuracy 0.92, precision 0.67, recall 0.592, and f1 score 0.63.

4.5 Discussion and Analysis

Table 4.1: Comparison between the models based on Accuracy, Precision, Recall, F1 Score

Model	Training Acc	Accuracy	Precision	Recall	F1 Score
ResNet with TL	0.98	0.96	0.95	0.96	0.96
ResNet	0.94	0.97	1.00	0.95	0.97
VGG19	0.95	0.46	0.46	1.00	0.63
InceptionV3 with TL	0.94	0.94	0.93	0.96	0.94
InceptionV3	0.94	0.94	0.925	0.973	0.948
Neural Network (NN)	0.92	0.92	0.67	0.592	0.63

The first two models with strong performance are ResNet and InceptionV3, particularly when using Transfer Learning (TL). Their balanced Precision, Recall, and F1 Score (ResNet with TL: Precision 0.95, Recall 0.96, F1 Score 0.96; InceptionV3 with TL: Precision 0.93, Recall 0.96, F1 Score 0.94) and high Accuracy (ResNet with TL: 0.96, InceptionV3 with TL: 0.94) demonstrate their effectiveness in accurately identifying and capturing relevant instances, making them dependable choices for a variety of tasks.However the comparison also shows that these indicators have trade-offs. As an example, VGG19 exhibits a high Recall (1.00) but a poor Precision (0.46), suggesting that although it successfully identifies the majority of important instances, it also incorrectly classifies a large number of irrelevant ones. In addition, this imbalance may be critical; for example, high Precision may be given priority when false positives have serious repercussions, while high Recall may be necessary when missing even one pertinent instance can be expensive (e.g., medical diagnosis).

Furthermore, the performance of the neural network (NN) model emphasizes how crucial model architecture and design are. Although its Accuracy of 0.92 is quite good, its Precision of 0.67 and Recall of 0.592when compared to CNN-based models like ResNet and InceptionV3 highlight the need of using specialist architectures, such as convolutional layers, for tasks like image classification.

The table presents a comprehensive evaluation of different models according to performance measures such as F1 score, recall, precision, testing accuracy, and training accuracy. A particular model is represented by each row, along with additional details like whether transfer learning (TL) was used. For instance, ResNet with transfer learning maintained a balanced precision,

recall, and F1 score values of 0.95, 0.96, and 0.96 respectively, and obtained a high training accuracy of 0.98 and a respectable testing accuracy of 0.96. In contrast, the F1 score of the VGG19 model was 0.63, indicating a major imbalance between precision and recall. The model also displayed a significantly lower testing accuracy of 0.46, along with precision and recall of 0.46 and 1.00. InceptionV3 showed consistent performance with and without transfer learning. The adaptability of the architecture was proved by InceptionV3, which showed constant performance across most measures both with and without transfer learning. Finally, the Neural network (NN) model performed a lower performance on all metrics, indicating that it was unable to handle the task's complexity. To sum up, this table offers a brief overview of the strengths and weakness of every model, therefore showing that ResNet and InceptionV3 are the best models for our study.

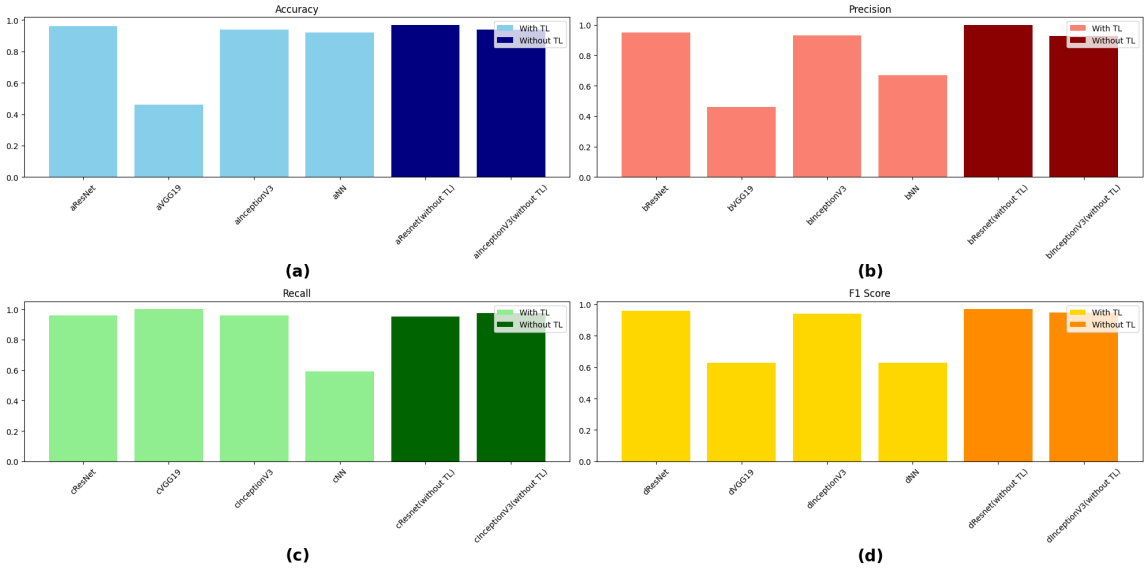


Figure 4.7: Displaying the results of all the models (a) accuracy , (b) Precision ,(c) Recall and (d) F1-Score.

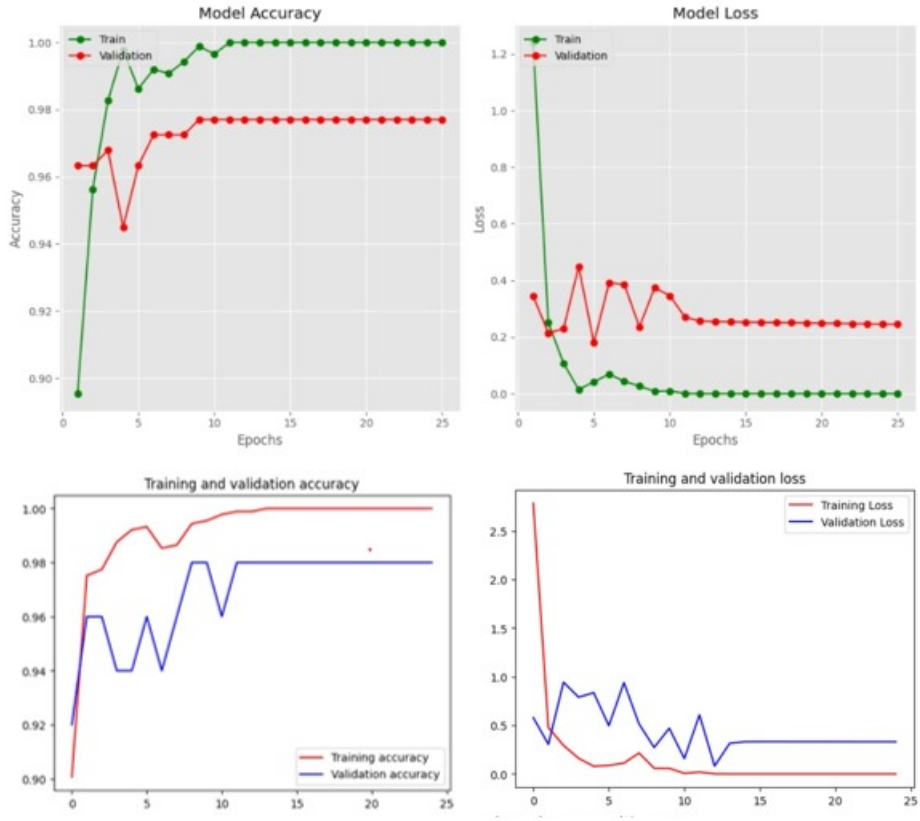


Figure 4.8: Displaying the model accuracy and loss. Prediction errors are quantified by loss, whereas accurate predictions are measured by accuracy. Overfitting can be found by keeping an eye on accuracy and loss throughout training and validation. Increasing training accuracy but decreasing validation accuracy or decreasing training loss but increasing validation loss are signs of overfitting. Techniques like regularization and adjusting hyperparameters can mitigate overfitting, with consideration for alternative metrics like precision and recall.

Chapter 5

Multi-Class Classification in Eye Disease Detection Using Deep Learning

5.1 Introduction

The detection of eye diseases, such as glaucoma, cataracts, and diabetic retinopathy, is critical for timely treatment and improved patient outcomes. The problem is inherently multi-class, requiring models to distinguish among several conditions. Deep learning, particularly convolutional neural networks (CNNs), has revolutionized the approach to image-based multi-class classification. This chapter delves into the theory of multi-class classification, the models used for image classification, and their comparative performance in eye disease detection tasks.

5.2 Multi-Class Classification in Deep Learning

In multi-class classification, an image is assigned to one of multiple predefined classes. For eye disease detection, this might involve classifying a retinal scan as "normal," "glaucoma," or "diabetic retinopathy." Deep learning models handle this complexity through specialized components such as the softmax activation function, which converts raw outputs into a probability distribution across classes. The class with the highest probability is chosen as the prediction.

The categorical cross-entropy loss function is commonly used to train models for such tasks. It penalizes incorrect predictions by comparing the predicted probabilities with the true class

labels. A critical aspect of evaluation involves metrics like accuracy, precision, recall, and the F1-score, which provide insights into model performance. For imbalanced datasets, weighted F1-scores or class-specific metrics are preferred to ensure that minority classes, often representing rarer diseases, are adequately represented [17] [18]

5.3 Models for Multi-Class Image Classification

Deep learning offers a variety of models for multi-class classification, each with strengths tailored to specific use cases. Convolutional Neural Networks (CNNs), such as ResNet and DenseNet, are the cornerstone of image classification. These networks learn spatial hierarchies of features, with ResNet employing skip connections to overcome vanishing gradients and DenseNet emphasizing feature reuse for efficiency.

Another approach is the use of Deep Decision Networks (DDNs), which cluster similar classes using confusion matrices and adopt hierarchical decision-making strategies. These networks reduce misclassifications in datasets where some classes share overlapping features. For instance, in medical imaging, conditions like diabetic retinopathy and macular degeneration may exhibit similar visual patterns.

Transfer learning has also been highly impactful. Pretrained models such as VGG16 and ResNet50 are fine-tuned on medical datasets, significantly reducing training time and enhancing performance when labeled data is scarce. Models augmented with attention mechanisms or hybrid loss functions further boost their robustness, allowing them to focus on disease-specific features, such as lesions in retinal scans. [17] [18]

5.3.1 Improved Resnet50

In this thesis, we improved the performance of Resnet50 by considering the following two approaches:

1. Using cross-validation technique: Cross-validation is an important approach in DL, and it refers to the evaluation of a model on an independent dataset. The main objective of cross-validation is to ensure that the model generalizes well to unseen data rather than just fitting the training dataset. In cross-validation, the dataset is divided into multiple

subsets known as ”folds”. Cross-validation techniques train the model on some folds while validating on others. These techniques include a number of evaluations, which give a better estimation of the model’s performance and help to reduce ”overfitting”. One common approach for cross-validation is K-Fold Cross-Validation, in which the dataset is divided into k folds. In our model, we used 5-Fold cross validation. The model is trained on k-1 (4 in our model) folds and validated on the remaining fold, which is repeated k (5 times in our model) times so that each fold acts as a validation set once. Using Focal Loss function: Focal accuracy is a metric derived from focal loss in an attempt to handle the challenges of class imbalance within classification tasks. Where there is a class that has significantly fewer representations compared to the other classes, a model will often have poor performance since its high accuracy can only be achieved by correctly predicting the majority class. Focal accuracy modifies the traditional way of calculating the accuracy to focus on the performance of the minority class by changing the way contributions to accuracy are evaluated, based on the confidence level of the predictions. This implies that the predictions made with high confidence, whether correct or wrong, will affect the score differently than those predicted with lower confidence. [17] [18] [19]

5.4 Comparative Analysis of Multi-Class Models

Model	Advantages	Disadvantages	Examples
ResNet	Handles vanishing gradients; robust for deep networks	Computationally intensive	Diabetic Retinopathy Detection
DenseNet	Efficient parameter usage; strong feature reuse	Overfitting on small datasets	Cataract Classification
Deep Decision Networks	Reduces inter-class confusion; hierarchical learning	Requires careful class clustering	Retinal Disease Differentiation
Transfer Learning	Fast convergence; leverages pre-learned features	May require extensive fine-tuning for medical data	General Eye Disease Detection

Figure 5.1: comparing different multi-class models

Comparative Analysis of Multi-Class Models When applied to eye disease detection, different models exhibit unique advantages and limitations. ResNet and DenseNet are strong baseline architectures due to their efficient feature extraction capabilities. However, DenseNet's extensive feature reuse can lead to overfitting, especially on small datasets, which are common in medical imaging. Deep Decision Networks, on the other hand, excel in reducing confusion among similar classes by employing hierarchical learning. While these networks offer higher accuracy in complex datasets, they require careful design and optimization of class clustering.

Transfer learning approaches stand out for their practicality. Pretrained models fine-tuned on domain-specific datasets like EyePACS (for diabetic retinopathy) or REFUGE (for glaucoma detection) achieve high accuracy with minimal data preprocessing. However, these models sometimes struggle with generalization when applied to new or unseen datasets.

A hybrid approach combining CNNs with attention mechanisms or advanced loss functions has shown significant promise. For instance, combining categorical cross-entropy with contrastive loss helps tackle inter-class similarity issues, leading to better performance in noisy datasets.

5.5 Applications in Eye Disease Detection

Multi-class classification has been effectively applied to various eye disease detection tasks. For diabetic retinopathy detection, CNNs like ResNet trained on EyePACS have achieved state-of-the-art performance by leveraging hybrid loss functions. Similarly, DenseNet has demonstrated excellent results in cataract classification, particularly when augmented with data augmentation techniques to enhance generalization.

In glaucoma detection, hierarchical models like DDNs have been instrumental in reducing inter-class confusion, ensuring accurate differentiation between early-stage glaucoma and normal eye scans. Transfer learning approaches, using models like VGG16 pretrained on ImageNet, have shown high accuracy across tasks with minimal training data. These successes underscore the adaptability of multi-class deep learning models in addressing diverse medical imaging challenges.

5.6 Conclusion

In this chapter, we explored the critical role of multi-class classification in the detection of eye diseases using deep learning techniques. We highlighted how models like Convolutional Neural Networks (CNNs), particularly ResNet, DenseNet, and Deep Decision Networks (DDNs), are adept at handling the complexities of distinguishing among various eye conditions such as glaucoma, cataracts, and diabetic retinopathy.

We emphasized the importance of specialized techniques, including the softmax activation function and categorical cross-entropy loss, which facilitate effective training and evaluation of these models. Moreover, we discussed innovative approaches like cross-validation and focal loss, which enhance model performance, particularly in addressing class imbalances.

Comparative analysis revealed that while traditional architectures like ResNet and DenseNet provide strong baselines, hybrid models leveraging attention mechanisms and advanced loss functions demonstrate significant promise in improving accuracy. Transfer learning, utilizing pretrained models on relevant datasets, further enhances model efficiency and accuracy with limited data.

Ultimately, the integration of multi-class classification techniques is vital for developing robust models capable of accurately detecting diverse eye diseases. This foundation sets the stage for future advancements in the application of these models in real-world healthcare scenarios, aiming to improve patient outcomes through timely and precise diagnosis.

Chapter 6

Analysis

6.1 Introduction

In this chapter, we will conduct a comprehensive comparison of our model against several prominent deep learning architectures, including ResNet50, the original code implementation, and VGG19. Our focus will be on multiclass classification of ocular diseases, where we aim to evaluate how each model performs in accurately identifying various conditions from fundus images. By analyzing metrics such as accuracy, precision, and recall, we will gain insights into the strengths and weaknesses of each architecture. This comparative study will not only highlight the effectiveness of our model but also contribute to a deeper understanding of how different neural network designs impact performance in medical image classification tasks.

6.2 ResNet50 Model

The first model utilizes a Convolutional Neural Network (CNN) based on the pre-trained ResNet50 architecture for the classification of eye diseases from retinal images. The approach begins with data preprocessing, where the dataset is filtered to identify specific eye conditions, including cataract, diabetes, glaucoma, hypertension, myopia, age-related macular degeneration, and other abnormalities. The images are then resized, augmented, and labeled to create a balanced multi-class dataset for training.

The model leverages ResNet50 as the backbone, which is a deep CNN that uses residual

connections to improve performance and address vanishing gradient problems. The top layer of ResNet50 is removed, and custom fully connected layers are added to adapt the model to the specific task. These layers include dense layers with ReLU activation, dropout for regularization, and batch normalization to stabilize training. The final output layer has 8 units corresponding to the 8 classes of eye conditions, using a softmax activation function to output probabilities for each class.

To address class imbalance, the model is trained using categorical cross-entropy loss, and the Adam optimizer with a learning rate of 0.01 is employed. The training process includes early stopping, which halts training if the validation accuracy does not improve for 10 consecutive epochs, preventing overfitting. During training, the model's performance is evaluated using metrics such as accuracy, confusion matrices, and classification reports, with the final goal of applying the model to real-world retinal images for disease detection.

After training for 100 epochs, the model achieves a test accuracy of 0.85. However, during training, the model shows a plateau in validation accuracy, where the accuracy reaches 0.48 at epoch 29, with no significant improvement in subsequent epochs. The model's performance is further validated using a confusion matrix, which reveals the classification errors across different disease categories. Additionally, the ROC curve analysis for each class shows the model's ability to distinguish between conditions, with each class having a corresponding AUC (Area Under Curve) score. A custom focal loss function is also considered to potentially improve performance on class-imbalanced datasets by focusing more on difficult-to-classify samples.

In conclusion, the model demonstrates robust performance in identifying various eye diseases from retinal images, providing a reliable tool for real-time detection and aiding in the diagnosis of these conditions.

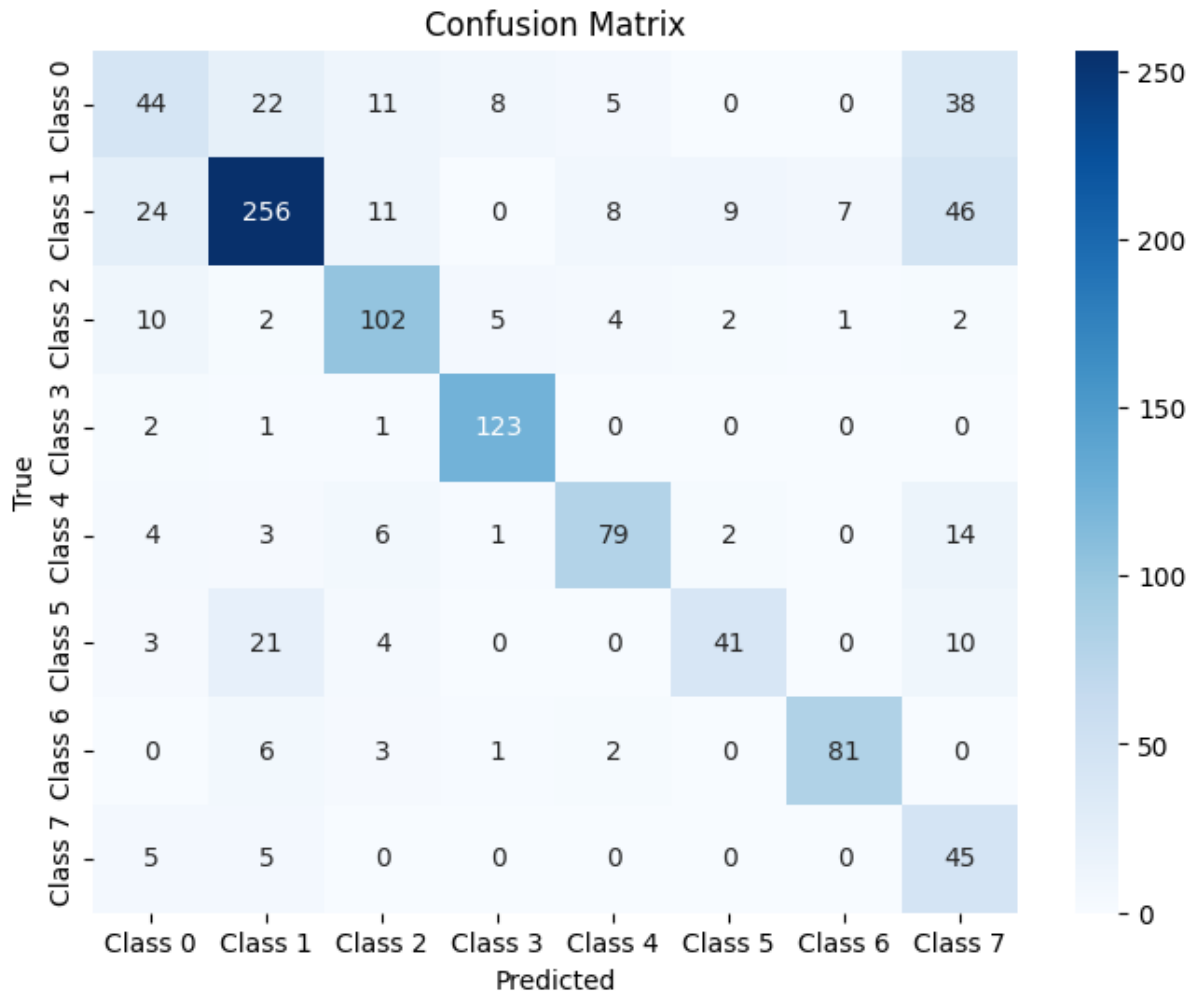


Figure 6.1: The confusion matrix for ResNet50 model.

The confusion matrix (figure 6.1) illustrates the performance of a classification model across eight classes. The diagonal values, particularly the highest count of 256 for Class 2, indicate a strong level of correct predictions. However, the off-diagonal elements reveal areas of misclassification, such as Class 0 being incorrectly predicted as Class 1 in 22 instances. Overall, the matrix highlights the model's strengths in accurately identifying certain classes while also pointing out the need for improvement in others, providing a comprehensive view of its classification capabilities.

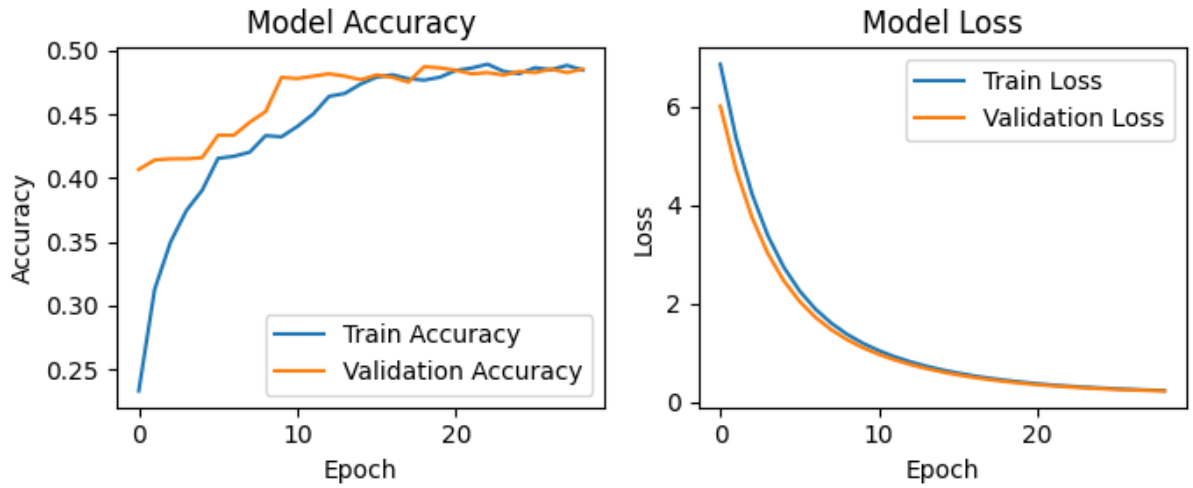


Figure 6.2: The Accuracy-Loss graph for ResNet50 model.

On the left of figure 6.2 , the accuracy plot shows both train accuracy (in blue) and validation accuracy (in orange) as the number of epochs increases. The train accuracy steadily improves, reaching around 0.5, while the validation accuracy exhibits fluctuations, suggesting potential overfitting as it does not consistently increase alongside the training accuracy.

On the right of figure 6.2, the loss plot displays the train loss (blue) and validation loss (orange). The train loss decreases significantly over the epochs, indicating effective learning, while the validation loss also trends downward, reflecting the model's ability to generalize. However, the gap between the two losses suggests that further tuning may be necessary to improve validation performance. Together, these plots provide insights into the model's learning dynamics and highlight areas for potential enhancement.

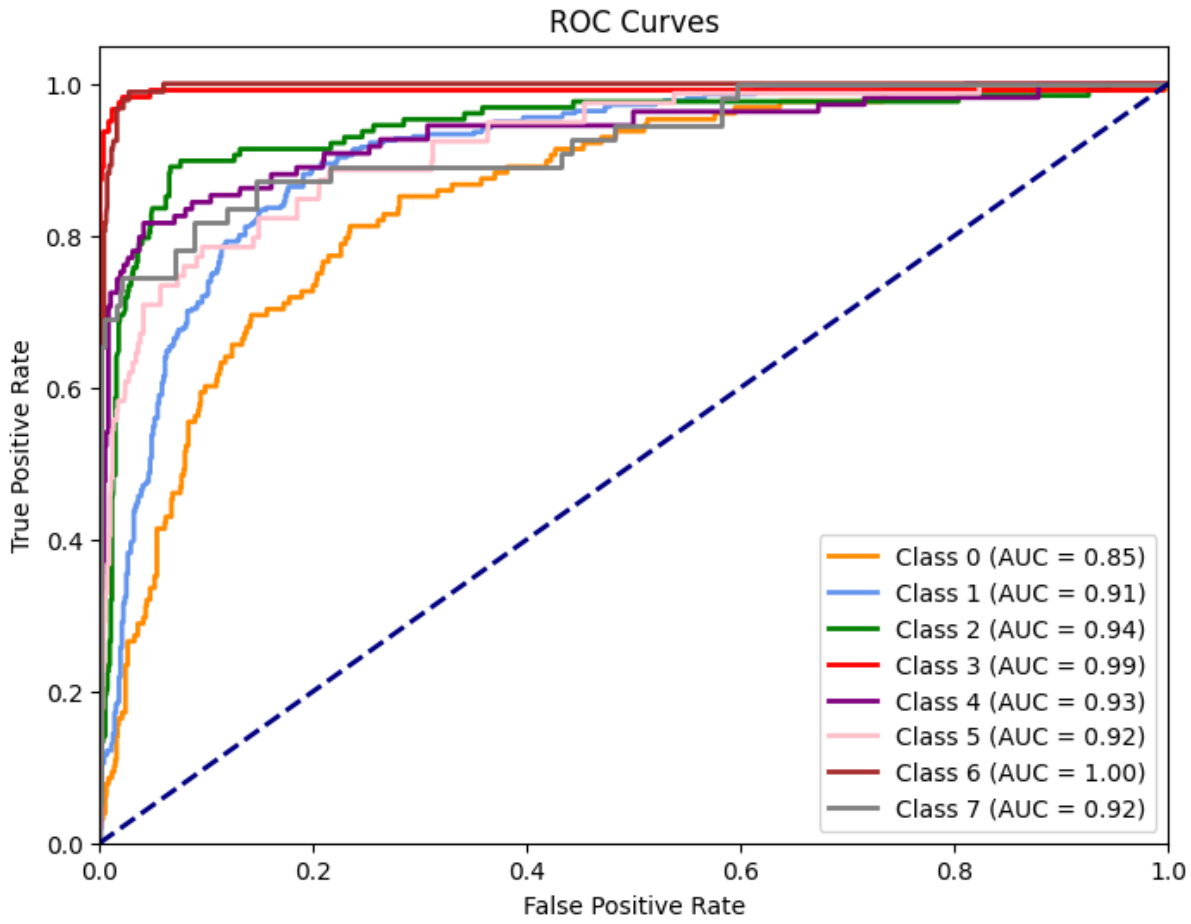


Figure 6.3: The ROC curves graph for ResNet50 model.

Figure 6.3 presents ROC curves for a ResNet50 model, illustrating the relationship between true positive rates and false positive rates for each class. Each curve is color-coded, with the corresponding area under the curve (AUC) values indicated in the legend. Notably, Class 6 and Class 7 achieve perfect AUC scores of 1.00, signifying exceptional model performance in distinguishing these classes. Class 3 also demonstrates strong performance with an AUC of 0.94. Other classes, such as Class 1 and Class 5, show robust AUC scores of 0.91 and 0.93, respectively. Overall, the curves approach the top-left corner of the plot, indicating high sensitivity and low false positive rates, thereby highlighting the model's effectiveness in multi-class classification and its ability to accurately identify various categories.

6.3 ResNet50 Multi-Class Model

The second deep learning model tested and used in our website implementation leverages the ResNet50 architecture, known for its residual learning mechanism, which helps in mitigating the vanishing gradient problem. By using a pre-trained version of ResNet50, the model benefits from transfer learning, utilizing features learned from large-scale datasets like ImageNet. The base model is modified by removing its fully connected top layers, allowing the inclusion of a custom head tailored to the specific classification task of eye diseases. This custom head comprises several layers designed for effective feature extraction and classification.

The model architecture starts with a flattening of the ResNet50 output followed by two dense layers. These layers include ReLU activations, which help in capturing non-linear relationships in the data. Dropout is applied to prevent overfitting, especially as the dataset could be prone to imbalance across different disease categories. Batch normalization is introduced to stabilize the learning process, improving the model's convergence speed and overall performance. The final layer consists of 8 units with a softmax activation function to output class probabilities, where each class corresponds to a different eye disease: Normal, Cataract, Diabetes, Glaucoma, Hypertension, Myopia, Age-related Macular Degeneration, and Other conditions.

For the loss function, focal loss is chosen over traditional categorical cross-entropy due to its ability to focus on hard-to-classify examples, especially useful for imbalanced datasets where certain classes might be underrepresented. The Adam optimizer with a learning rate of 0.001 is utilized to ensure efficient optimization of the model parameters. During training, early stopping is employed to halt training when the validation performance ceases to improve, avoiding overfitting and unnecessary computational expense. Model checkpointing saves the best performing model during training, allowing for the restoration of the best model later.

The model's performance was evaluated at each epoch, with the training accuracy reaching 0.94 by epoch 20. However, the validation accuracy plateaued at 0.80, triggering early stopping at epoch 20, as the validation accuracy failed to improve beyond 0.81. This decision ensured the model did not overfit the training data.

Data preprocessing is a crucial part of this pipeline. Images are resized to 224x224 pixels

to be compatible with the ResNet50 input size, and data augmentation techniques are applied to further generalize the model, improving its robustness. After training, the model is evaluated on test data using various metrics, including accuracy, confusion matrix, classification report, and ROC curves. These metrics provide a comprehensive view of the model's performance, highlighting both the overall accuracy and its ability to distinguish between different disease categories.

Finally, the trained model is capable of classifying new retinal images, predicting the most likely disease category based on the image features. The output is a prediction of the disease class with the highest probability, aiding in early diagnosis and contributing to more accurate decision-making in clinical settings.

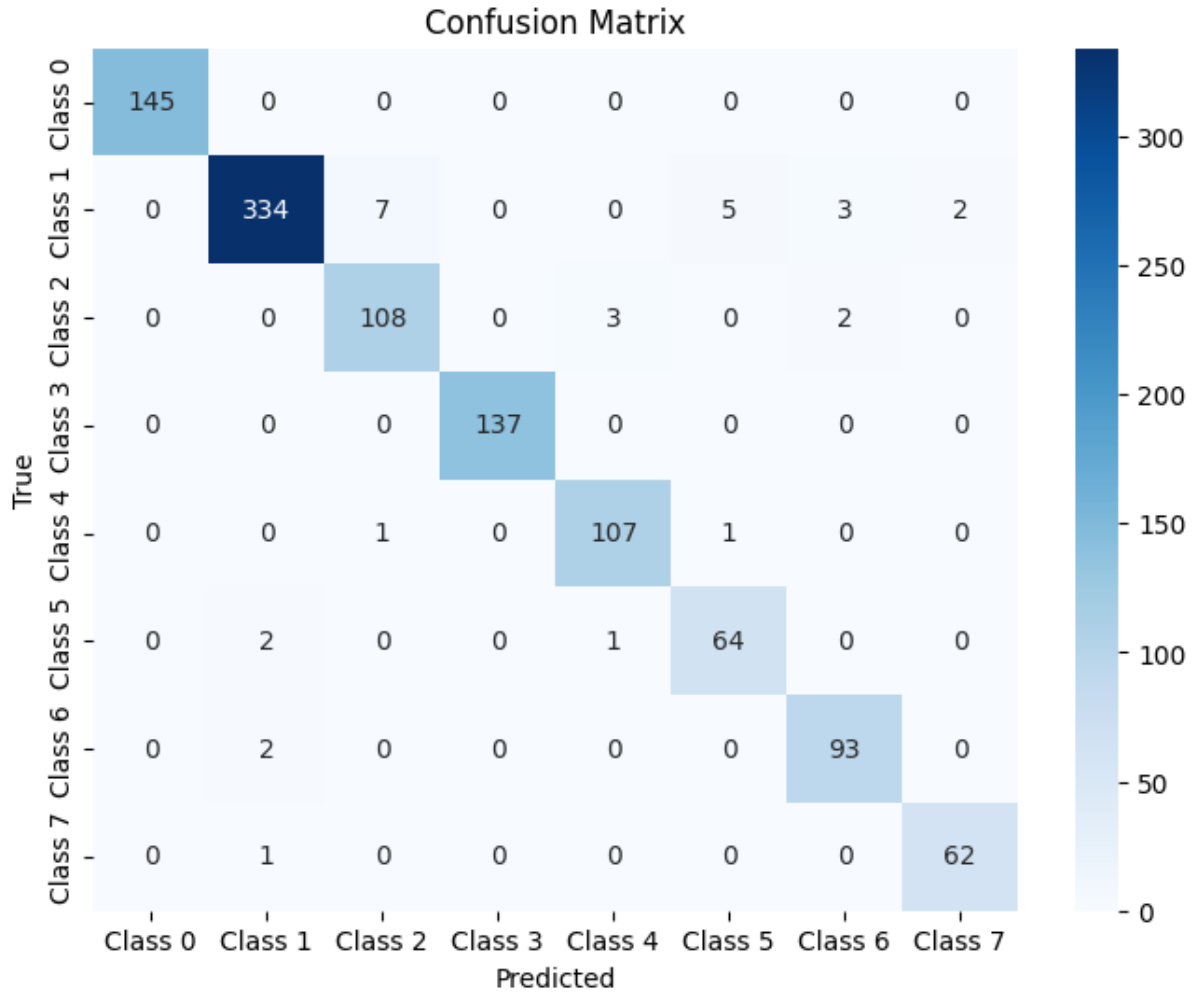


Figure 6.4: The confusion matrix for ResNet50 multi-class model.

Figure 6.4 displays a confusion matrix for a multi-class classification model, presenting

the true versus predicted classifications for each class. The matrix is structured with actual classes represented in rows and predicted classes in columns. Each cell indicates the number of instances, with darker shades signifying higher values. For instance, Class 1 shows a strong performance with 334 correct predictions, while Class 2 has 108 correct predictions but also misclassifications, as indicated by the off-diagonal entries. Class 6 and Class 7 also perform well, with 93 and 62 correct predictions, respectively. However, some classes exhibit misclassification, as seen with Class 0, which has 145 true positives but also multiple errors in predicting other classes. Overall, the confusion matrix effectively highlights the model's strengths and weaknesses in classifying different categories, providing insights into areas for potential improvement.

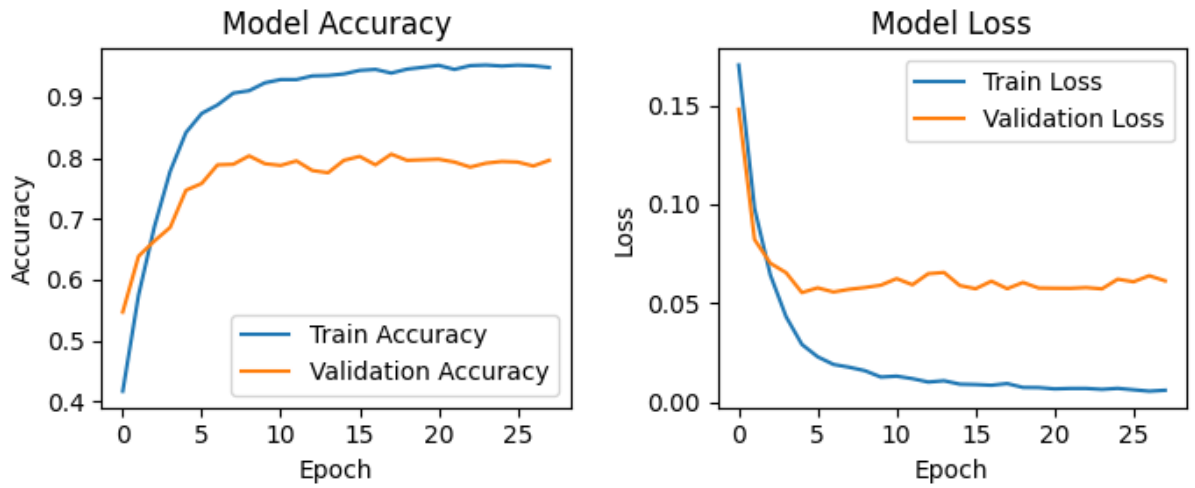


Figure 6.5: The Accuracy-Loss graph for ResNet50 multi-class model.

Figure 6.5 features two plots that evaluate the performance of ResNet Multi-Class classification model over training epochs, illustrating model accuracy and loss. The left plot shows model accuracy with the x-axis representing epochs and the y-axis indicating accuracy values from 0 to 1. The blue curve, depicting training accuracy, rises steadily from around 0.4 to approximately 0.9, indicating effective learning from the training data. In contrast, the orange curve for validation accuracy also increases but plateaus around 0.8, suggesting some overfitting as the model performs better on training data than on unseen data. The right plot illustrates model loss, where the blue curve for training loss decreases sharply from higher values to about 0.05, indicating a reduction in prediction errors. Meanwhile, the orange curve for validation loss

also drops but remains higher than the training loss, highlighting inconsistencies in the model's performance on validation data. Together, these plots provide insights into the model's training dynamics, emphasizing the need for a balance between training and validation performance to enhance generalization. Figure 6.6 presents a Receiver Operating Characteristic (ROC) curve

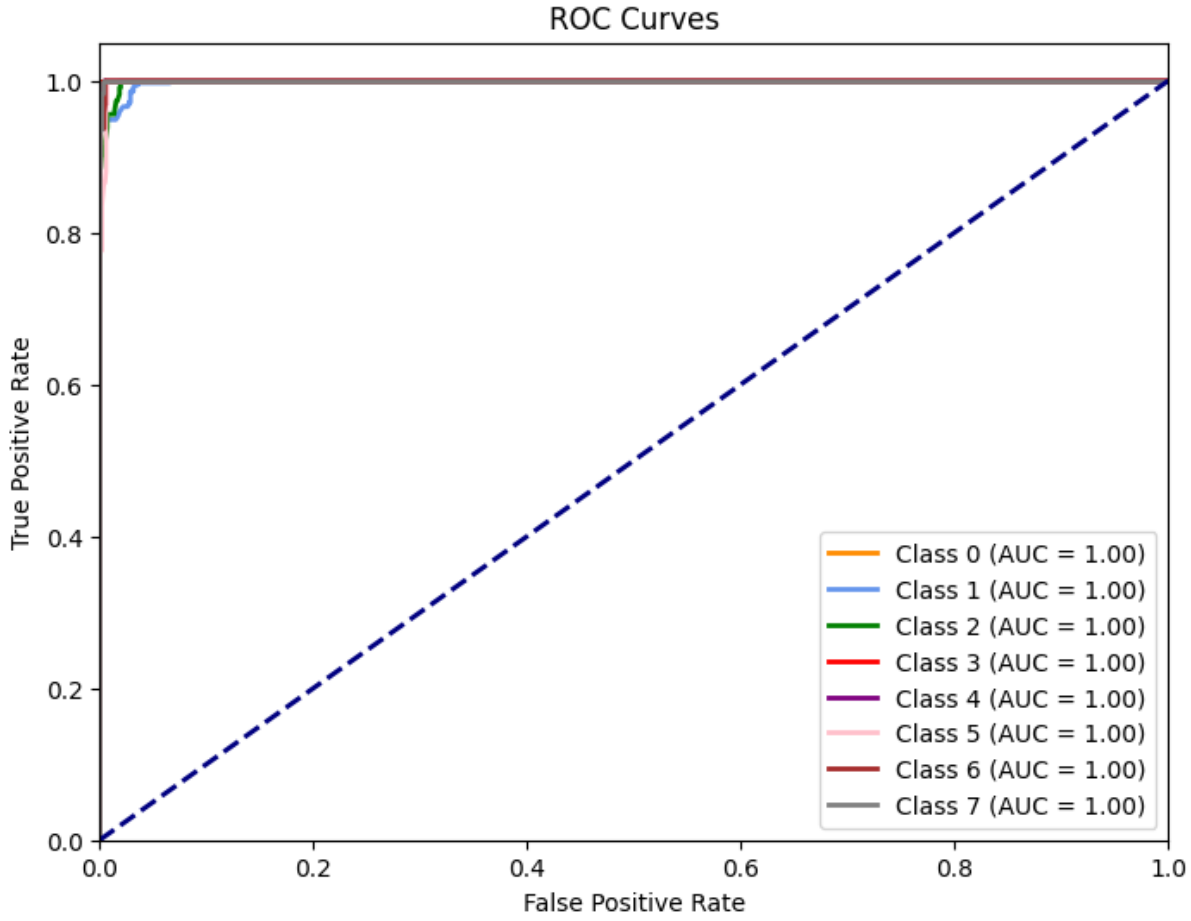


Figure 6.6: The ROC curve graph of ResNet50 multi-class model.

plot for a ResNet multi-class classification model, showcasing the model's performance across different classes. The x-axis represents the false positive rate (FPR), ranging from 0 to 1, while the y-axis indicates the true positive rate (TPR), also from 0 to 1. Each curve corresponds to a different class, with color-coded lines representing their respective performance. Notably, all classes have an Area Under the Curve (AUC) of 1.00, indicating perfect classification performance. The curves are close to the top-left corner of the plot, demonstrating high true positive rates with minimal false positives. The dashed diagonal line represents the baseline performance of a random classifier, which the model significantly outperforms. Overall, this ROC

curve plot effectively illustrates the model's exceptional ability to distinguish between classes, confirming its reliability and effectiveness in classification tasks.

6.4 VGG-19 Multi-Class Model

The last model we employed for analysis is a Convolutional Neural Network (CNN) based on the pre-trained VGG19 architecture, designed for classifying eye diseases from retinal images. The analysis begins with dataset preparation, where data is loaded from a CSV file and filtered to focus on relevant eye conditions. Visualizations of gender and age distributions are created, and new features are engineered to represent various eye conditions. The images undergo preprocessing, which includes resizing and applying augmentation techniques to enhance model generalization.

The core of the model leverages the VGG19 architecture, which is fine-tuned by adding fully connected layers tailored for specific classification tasks related to diseases such as cataract, diabetes, and glaucoma. To address any class imbalance in the dataset, class weights are computed, ensuring that the model does not favor overrepresented classes. The training process utilizes augmented data and incorporates early stopping to mitigate overfitting.

Training results demonstrate performance over 100 epochs, with notable metrics: at Epoch 99, the training accuracy is 0.90 and validation accuracy is 0.77 ; at Epoch 100, training accuracy reaches 0.90 while validation accuracy increases to 78.70

Upon completion of training, the model's performance is evaluated using metrics such as accuracy, confusion matrices, and classification reports. Ultimately, the trained model can be applied to new retinal images for real-time detection of eye diseases, providing diagnostic predictions based on the patterns it has learned.

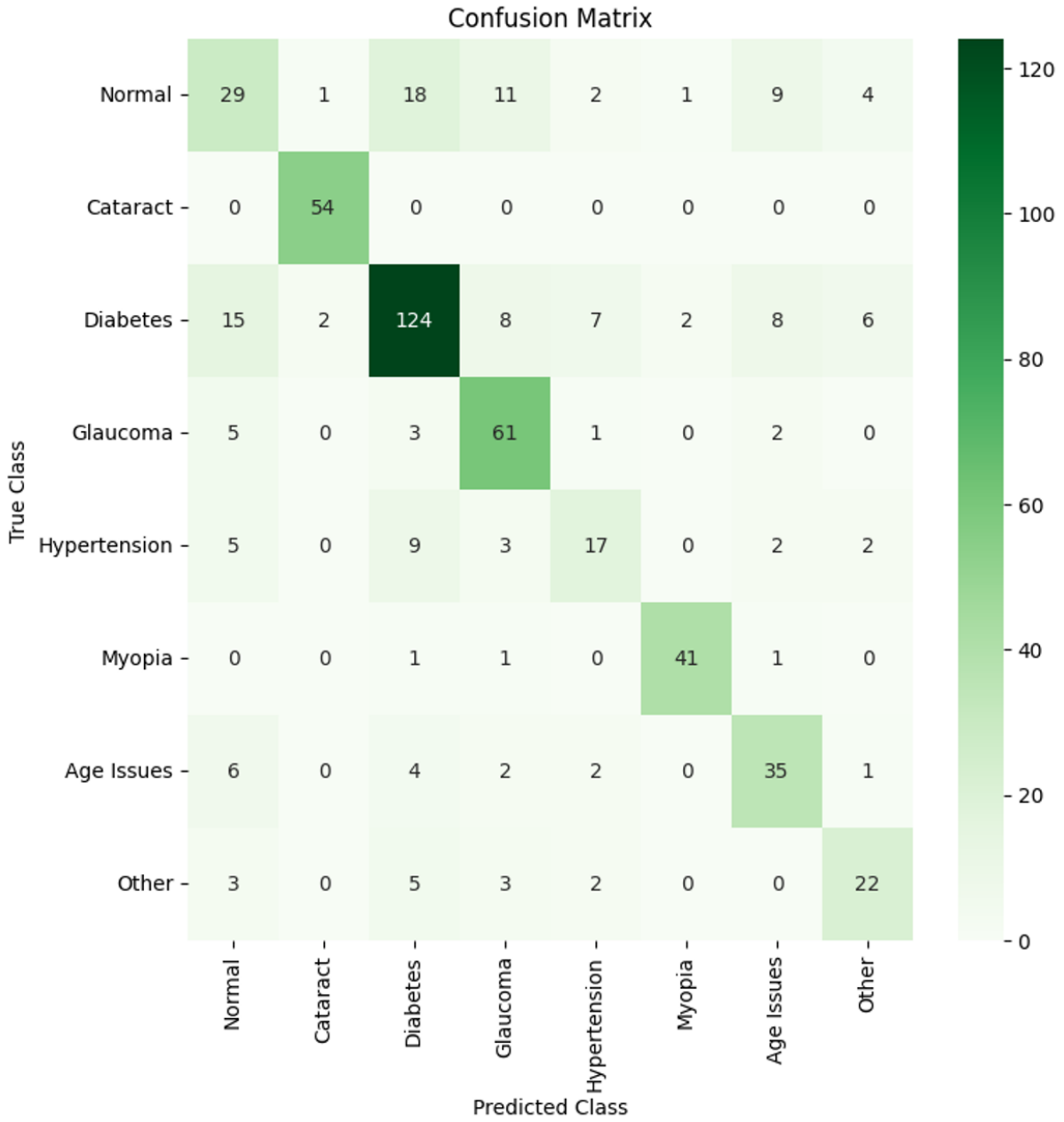


Figure 6.7: The confusion matrix for VGG19 multi-class model.

Figure 6.7 features a confusion matrix that evaluates the performance of a classification model across various medical conditions, with true classes represented on the y-axis and predicted classes on the x-axis. Each cell indicates the number of instances classified into each category. For example, the model accurately predicts 29 instances as ”Normal,” misclassifying 1 as ”Cataract” and 18 as ”Diabetes.” In the case of ”Cataract,” it correctly identifies 54 instances with no false positives, while ”Diabetes” shows 124 correct predictions but 15 in-

stances misclassified as ”Glaucoma.” The model correctly predicts 61 cases of ”Glaucoma,” although some are misclassified into other categories. Accurate predictions for ”Hypertension” are noted, but there are misclassifications into ”Age Issues.” The categories of ”Myopia” and ”Age Issues” show fewer instances, with some misclassifications as well. The gradient color scale in the matrix highlights these performance metrics, with darker shades indicating higher counts, thereby visually emphasizing the model’s strengths and weaknesses. This comprehensive overview facilitates an assessment of the model’s accuracy and identifies areas needing improvement.

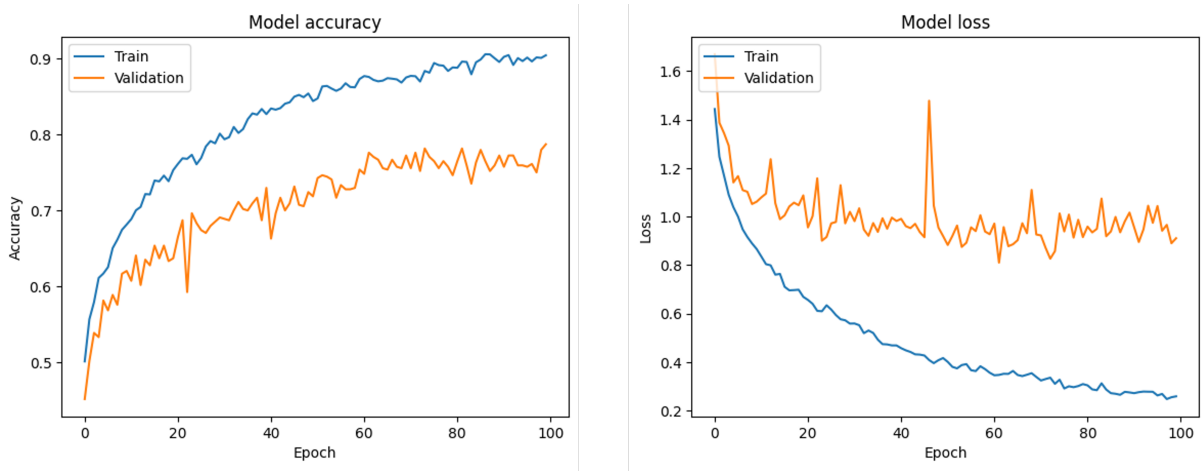


Figure 6.8: The Accuracy-Loss graph for VGG19 multi-class model.

Figure 6.8 presents two plots illustrating the performance of a Convolutional Neural Network (CNN) over 100 epochs, focusing on model accuracy and model loss. The left plot displays the training accuracy (in blue) and validation accuracy (in orange) throughout the epochs. Training accuracy steadily increases, reaching around 0.90 by the end of the training, while validation accuracy also improves but exhibits slight fluctuations, concluding at approximately 0.78 . This variability indicates some inconsistency in the model's performance on unseen data. In the right plot, the training loss (in blue) and validation loss (in orange) are depicted across the same epochs. The training loss decreases consistently, suggesting effective learning by the model. Conversely, the validation loss shows fluctuations and a slight upward trend at certain points, which may indicate overfitting as training progresses. Together, these plots provide a visual representation of the model's training process, highlighting the relationship between accuracy and loss, and offering insights into the model's learning behavior and generalization capabilities.

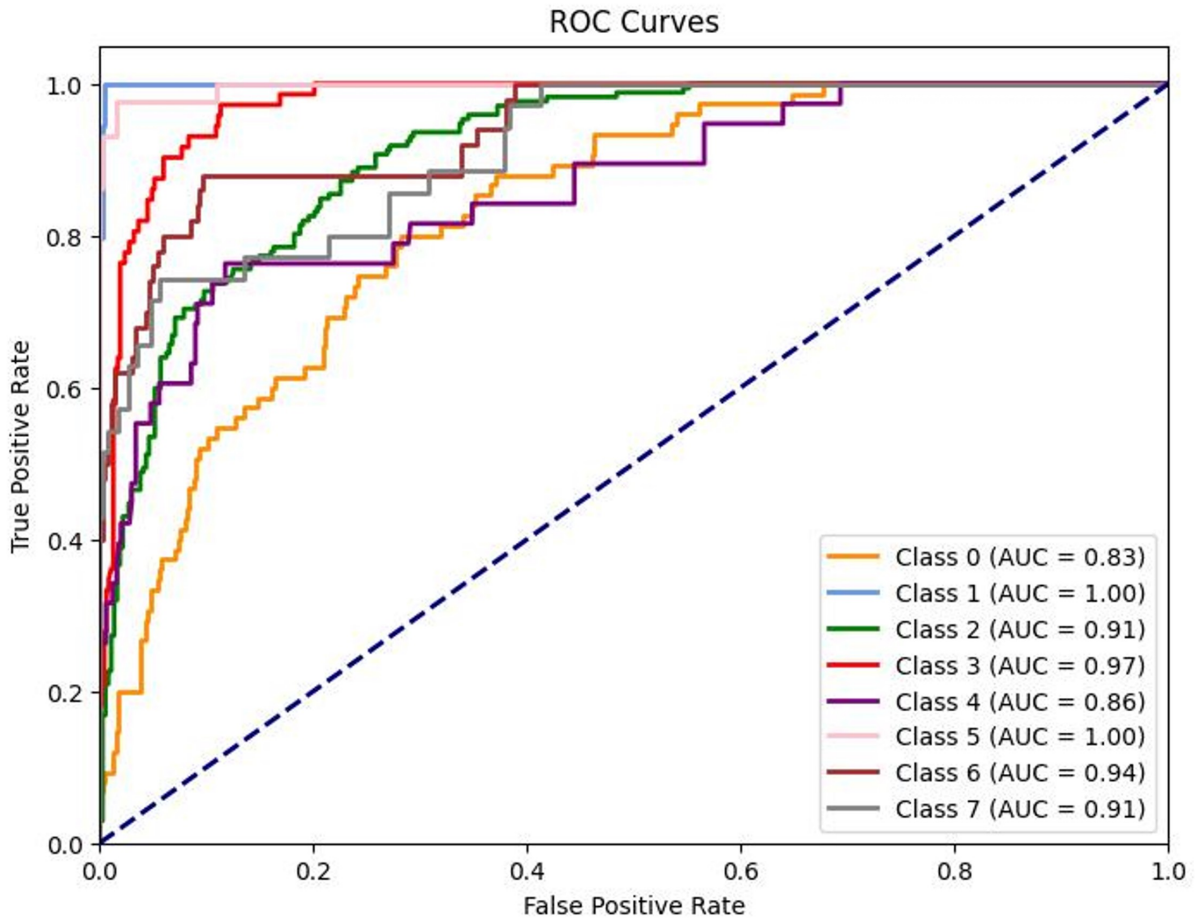


Figure 6.9: The ROC graph for VGG19 multi-class model.

Figure 6.9 displays a set of Receiver Operating Characteristic (ROC) curves for multiple classes in a classification problem, illustrating the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity). Each colored curve represents a different class, with the diagonal dotted line indicating the performance of a random classifier; curves above this line signify better-than-random performance. The Area Under the Curve (AUC) values for each class are noted as follows: Class 0 has an AUC of 0.83 (orange), Class 1 achieves a perfect AUC of 1.00 (red), Class 2 has an AUC of 0.91 (gray), Class 3 scores 0.97 (green), Class 4 has an AUC of 0.86 (blue), Class 5 and Class 6 both reach an AUC of 0.94 (purple and brown, respectively), and Class 7 scores 0.91 (black). These curves demonstrate varying performance across classes, with Class 1 achieving perfect classification and other classes displaying high accuracy as indicated by their AUC values. Overall, the ROC curves provide a comprehensive view of the model's ability to distinguish between different classes.

6.5 Conclusion

The ResNet50 Multi-Class Model (Second) is the best-performing model, achieving the highest training accuracy (0.94) and effectively addressing class imbalance with focal loss. While its validation accuracy plateaus early, it still outperforms the other models in terms of overall accuracy and robustness. Further fine-tuning could improve its ability to generalize to unseen data, making it the most reliable option for eye disease classification from retinal images.

Chapter 7

Implementation

7.1 Introduction

In the modern convergence of technology and healthcare, developing applications that improve diagnostic capabilities is crucial. This section focuses on creating an innovative app that detects eye diseases, a vital aspect of maintaining ocular health. With the growing prevalence of vision-related disorders worldwide, timely and accurate detection can have a profound impact on patient outcomes.

Building on our junior project's foundational model, this app combines advanced machine learning algorithms and image processing techniques to analyze ocular images. It provides doctors with immediate insights into potential eye conditions. By integrating user-friendly interfaces and robust analytical tools, the app aims not only to facilitate early diagnosis but also to educate users about eye health.

7.2 Prototype

The Eye Scan application serves as an intuitive mobile platform focused on the early detection and management of ocular diseases, designed for healthcare professionals. It features a straightforward interface that simplifies navigation, alongside essential functionalities such as uploading high-quality fundus images for analysis and employing deep learning for disease identification. The app will also present the results along with comprehensive details regarding

the signs and symptoms, highlighting key indicators of the condition. Additional capabilities include user profiles to monitor eye health history, and adjustable text sizes and alternative text for images, catering to diverse user needs. The Figma design enhances this experience with a modern UI, incorporating a bottom navigation bar for easy access to the following sections: Home, Upload, and Settings. The results screen clearly presents findings and recommendations. Engaging content provides information on ocular diseases, and the user profile allows for tracking health history with visual aids. Overall, the Eye Scan application aims to improve accessibility and efficiency in ocular healthcare, empowering users to take proactive steps in managing their eye health.

Figma's eye scan app has a simple, straightforward, user-friendly design that emphasizes clarity and simplicity (shown in figure 7.1) . The welcome screen features a prominent eye logo with a slogan placeholder, a green "Login" button, and options for Google sign-up or new user registration, set against a white background with green and blue accents to convey trust and medical precision. he image capture screen offers clear option for selecting from the gallery, ensuring ease of use. The results screen displays a side-by-side comparison of normal and diabetic retinopathy retinas, with a brief explanation of the condition and its symptoms, maintaining a straightforward and informative design. Overall, the app combines modern aesthetics with functionality, making it easy for users to navigate and understand.

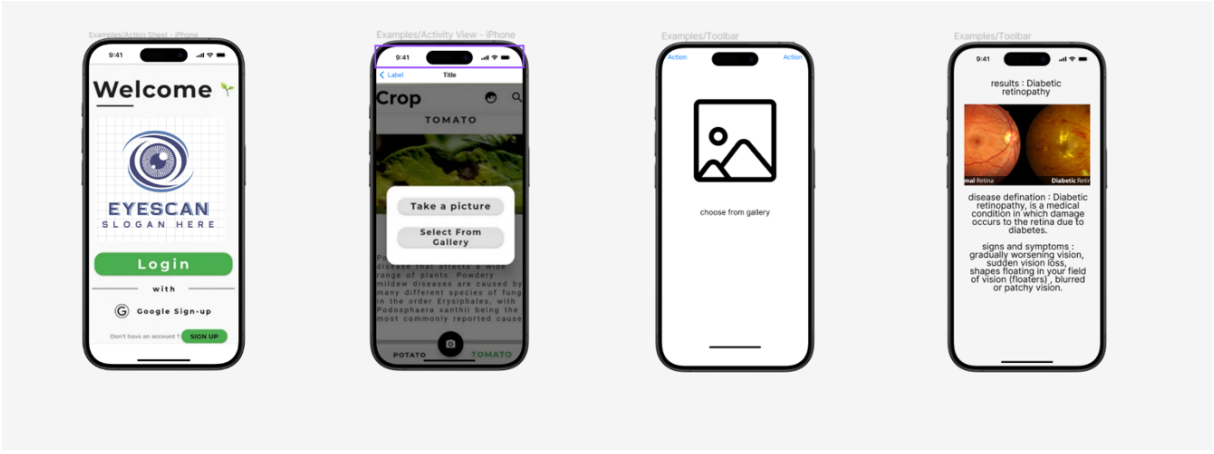


Figure 7.1: Prototype using Figma .

7.3 System Development

The Eye Scan application is designed with a user-centric approach, focusing on simplicity, functionality, and aesthetic appeal. The overall design follows a modern and clean layout, utilizing a soothing color palette that minimizes eye strain, enhancing the user experience during prolonged use.

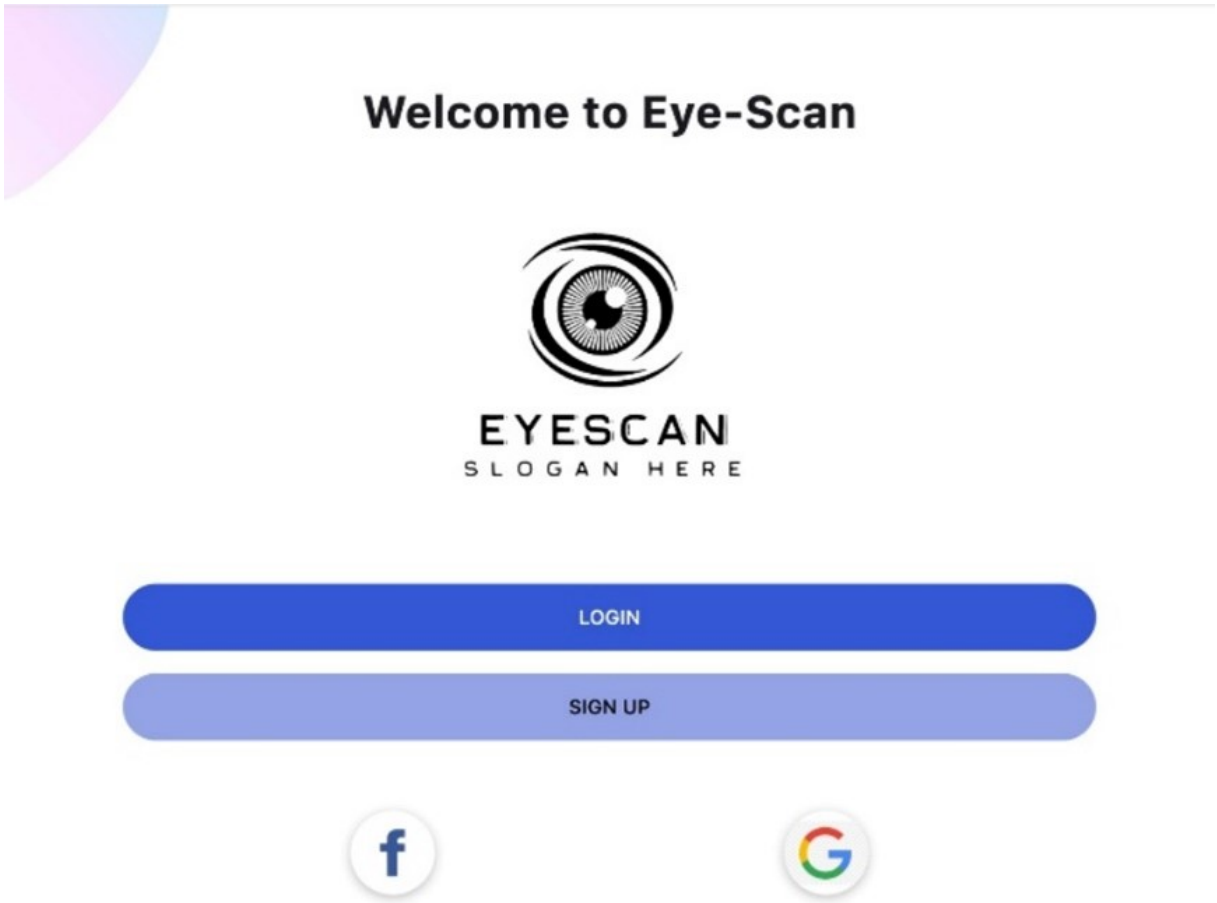


Figure 7.2: Login page .

In figure 7.2 , once you open the Eye-Scan application for the first time, you are greeted by the login page, which sets the stage for accessing important eye health services. The message ”Welcome to Eye-Scan” prominently displayed at the top indicates that this app is dedicated to detecting eye diseases, emphasizing its focus on eye care.

Below the welcome message, you’ll see a stylized eye logo that symbolizes the app’s mission. If you already have an account, you can click the blue ”LOGIN” button to access your

profile and begin using the diagnostic features. If you're new to the app, the "SIGN UP" button allows you to create an account easily, ensuring you can start monitoring your eye health without delay.

Additionally, the page offers the option to sign in using your Facebook or Google accounts, making the process quick and convenient. The clean, modern design, featuring a blue and white color scheme, enhances usability and reflects a professional approach to health technology. Overall, this login page is your gateway to vital tools for detecting and managing eye diseases effectively.

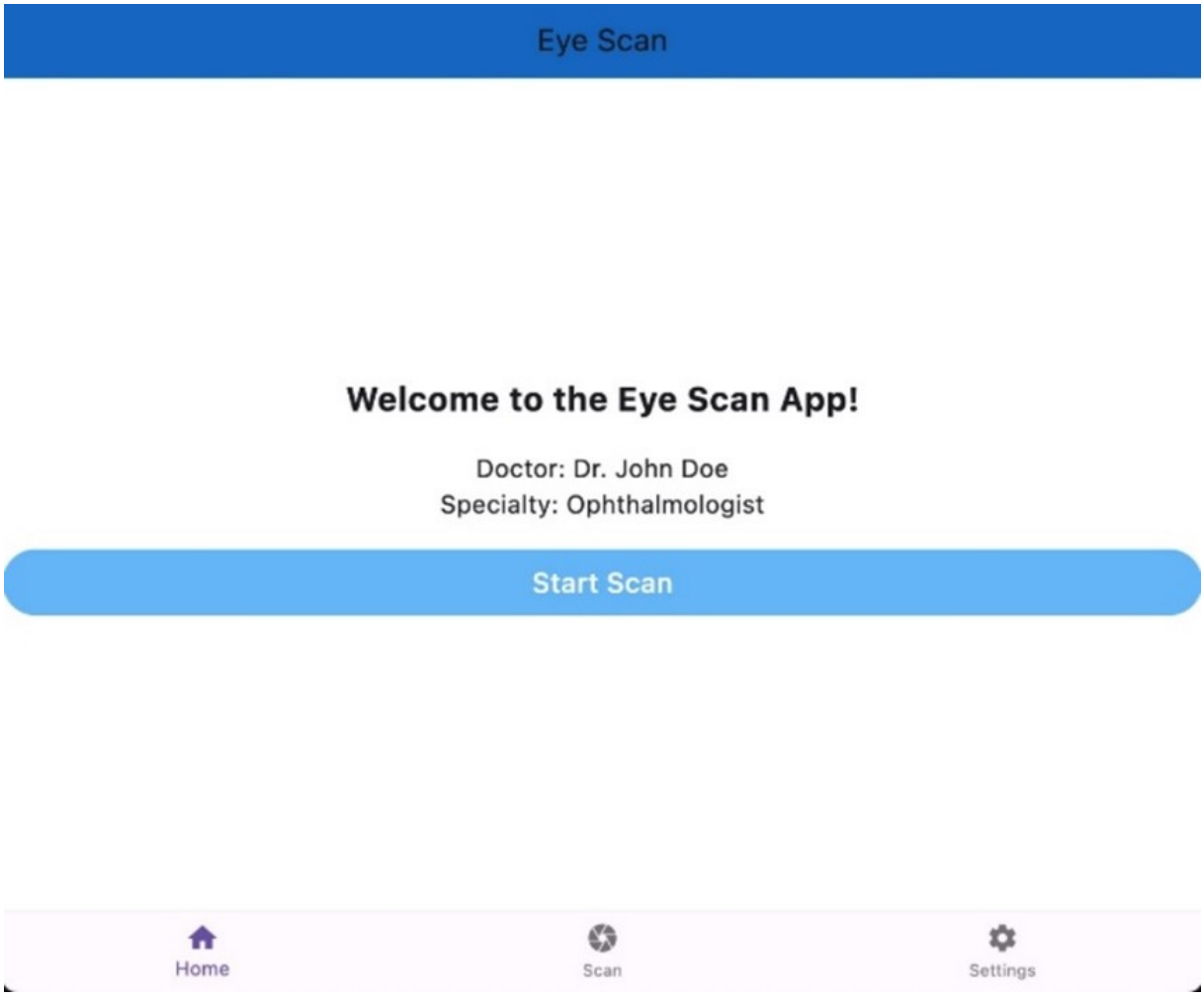


Figure 7.3: Home page .

As you navigate to the main interface of the Eye Scan application (figure 7.3), you are greeted by a clear and professional layout that welcomes you to the app. The message "Welcome to the Eye Scan App!" indicates that you are now ready to utilize its features for detecting

eye diseases.

Beneath this welcome message, your role as a user is acknowledged, reinforcing the app's focus on eye health. In the center of the screen, the prominent "Start Scan" button invites you to begin the scanning process, making it easy to initiate assessments quickly.

At the bottom of the screen, you'll find navigation options labeled "Home," "Scan," and "Settings." This layout provides easy access to different sections of the app, allowing you to manage patient scans, review results, and adjust settings as needed.

Overall, this interface is designed to be intuitive and effective, equipping you with the tools necessary to conduct eye health assessments and deliver quality care.

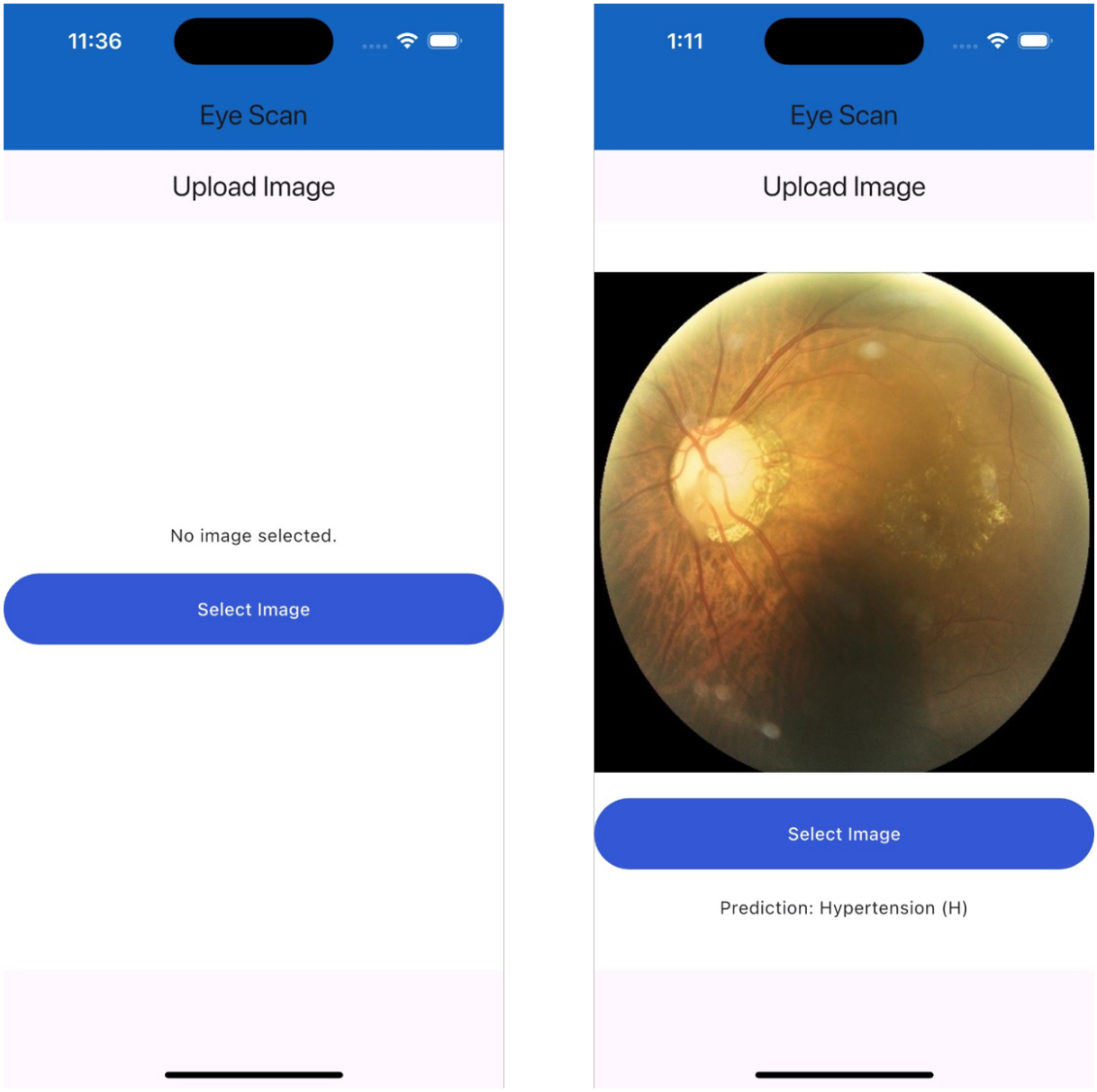


Figure 7.4: upload image page .

Figure 7.4 , the Upload Image page of the EyeScan app features two distinct states. In the first state, when no image is selected, the page displays a message indicating ”No image selected” beneath the title. A prominent ”Select Image” button invites users to upload an image from their device, encouraging engagement.

In the second state, once an image is uploaded, the page displays the retinal image prominently, showcasing the details for user reference. Below the image, the prediction result appears, such as ”Prediction: Hypertension (H),” clearly indicating the identified condition and its classification. This design effectively guides users through the process of image submission and

provides immediate feedback on their eye health.

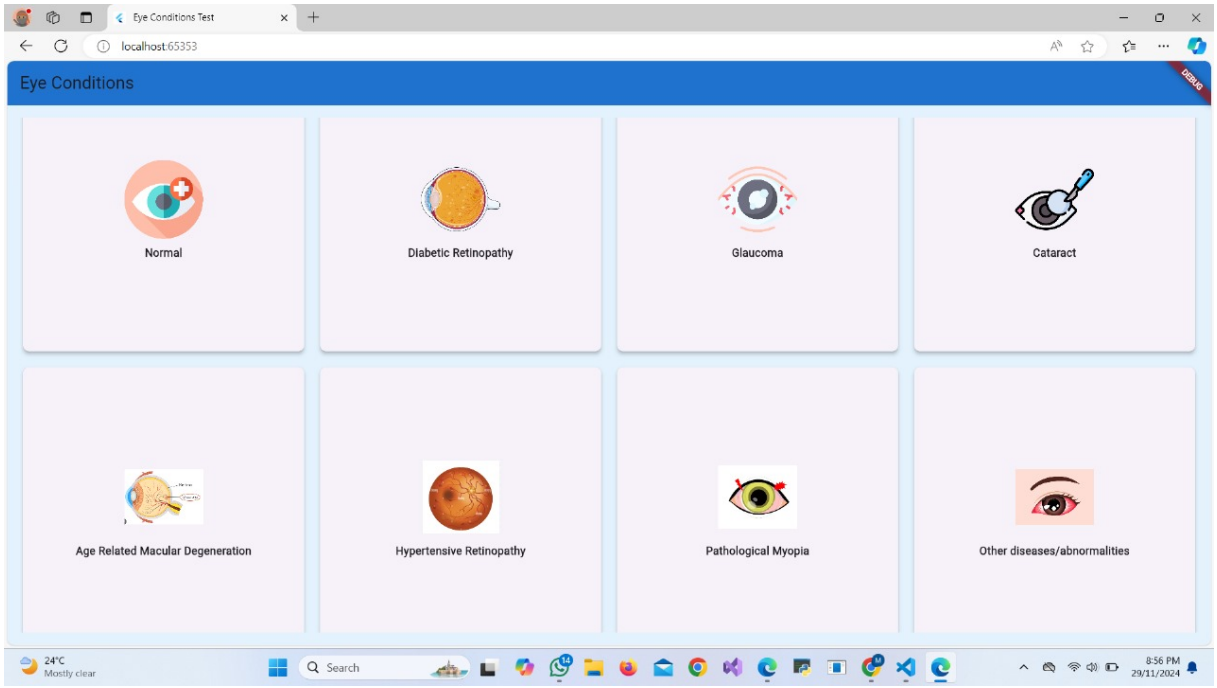


Figure 7.5: Diseases page .

After the doctor completes the scanning process, they should navigate to the disease page (figure 7.5) to learn more about various eye conditions. This page features a visually organized grid layout that highlights common eye diseases, with each condition represented by an icon or image for easy identification. Categories include ”Normal,” indicating healthy eyes, as well as prevalent conditions like Diabetic Retinopathy, which affects the retina due to diabetes, and Glaucoma, associated with increased eye pressure that damages the optic nerve. Other entries include Cataracts, known for clouding the lens; Age-Related Macular Degeneration, which impacts central vision; and Hypertensive Retinopathy, resulting from high blood pressure. Additionally, the page features Pathological Myopia, a severe form of nearsightedness, and a section for ”Other Diseases/Abnormalities,” covering various additional eye conditions. This layout not only enhances accessibility but also provides doctors with essential information to aid in diagnosis and patient care.

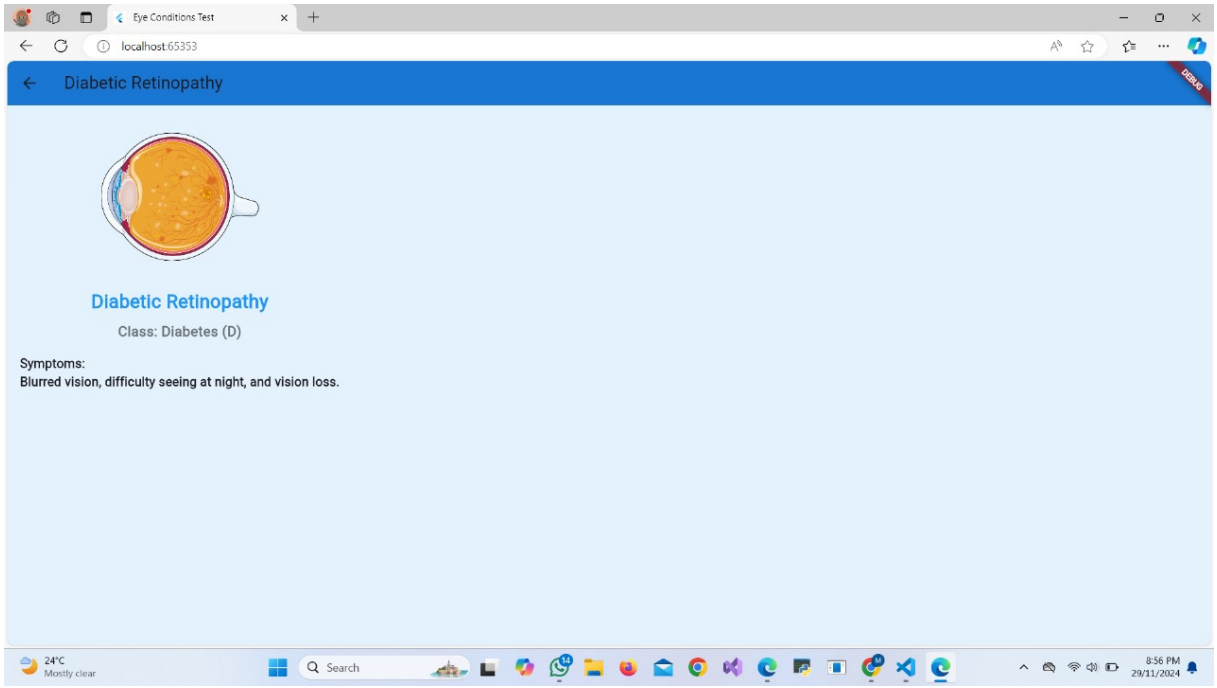


Figure 7.6: Diseases informations page .

After the doctor selects a specific condition(figure 7.6) , such as Diabetic Retinopathy, they are directed to a detailed information page that provides comprehensive insights into the disease. This page includes a description of Diabetic Retinopathy as a complication of diabetes that affects the blood vessels in the retina, potentially leading to severe vision impairment or blindness if untreated. Key symptoms highlighted are blurry vision, which may fluctuate, significant vision loss in advanced stages, difficulty seeing at night, and the presence of floaters—dark spots or shapes in the visual field. Additionally, the page may outline causes, such as prolonged high blood sugar levels damaging retinal blood vessels, and risk factors like the duration of diabetes and poor blood sugar control. Treatment options, including laser therapy and diabetes management, are also discussed. This structured layout ensures that doctors can quickly access essential information, facilitating effective diagnosis and patient communication.

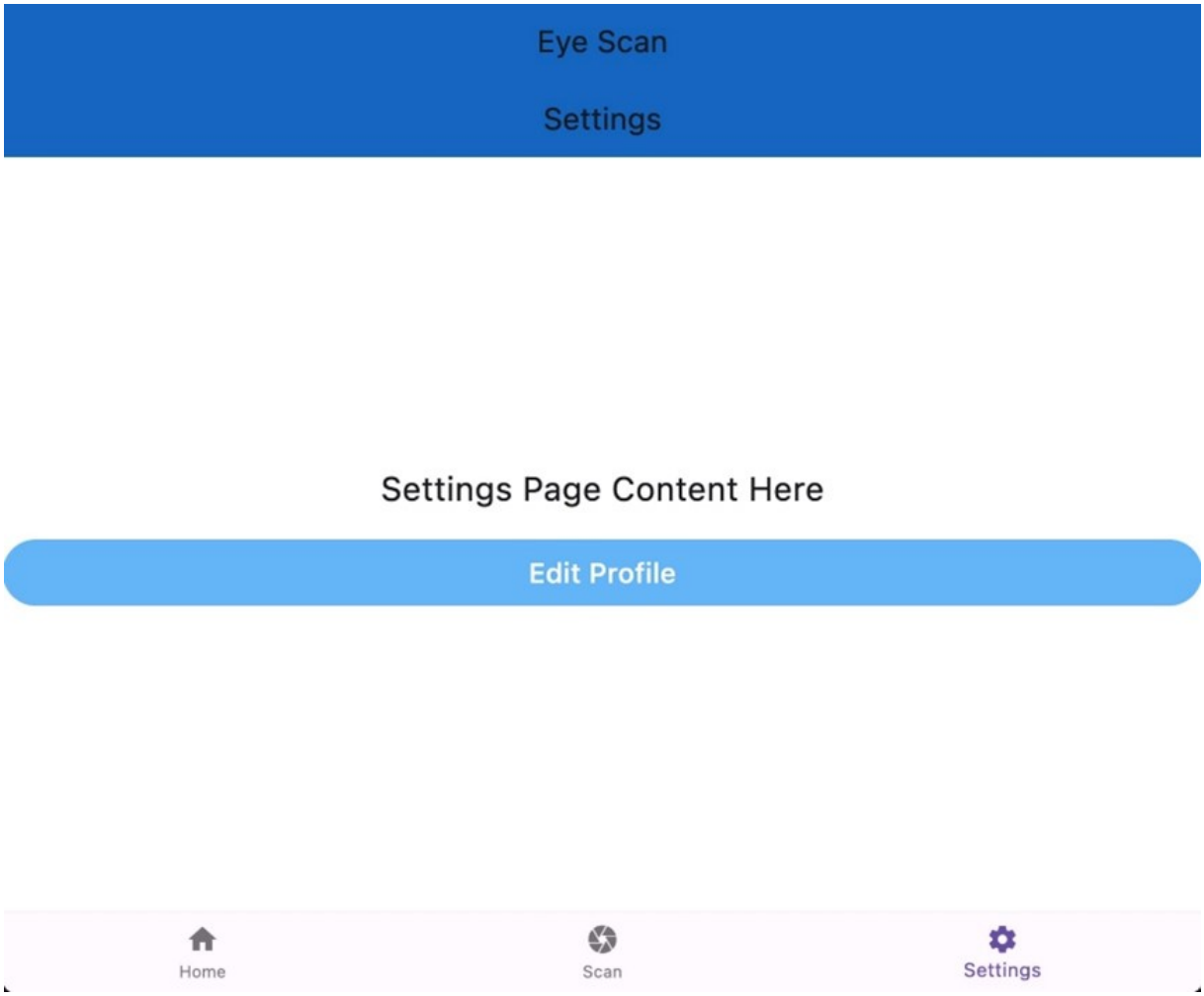


Figure 7.7: Settings page .

In figure 7.7 as you access the Settings page of the Eye Scan application, you are presented with a clean and straightforward layout. At the top, the app's title, "Eye Scan," remains visible, ensuring you are oriented within the application.

In the center of the screen, a message reads "Settings Page Content Here," indicating that this section is dedicated to managing your preferences and account details. The prominent "Edit Profile" button invites you to make any necessary changes to your profile information, enhancing your experience with the app.

7.4 EyeScan Web Page

To simplify the process for doctors and enhance accessibility, we developed a user-friendly web page powered by deep learning. This platform allows medical professionals to directly upload patient data or eye images for analysis. Utilizing advanced deep learning algorithms, the system efficiently detects and classifies eye diseases with high accuracy. The intuitive interface ensures that doctors can make quick, reliable diagnoses without needing specialized tools, streamlining the overall diagnostic process and improving patient care.

Figure 7.8: Eye Disease Classifier home page.

Figure 7.8 displays a user interface for a ”Doctor’s Dashboard.” The layout includes the doctor’s name, ”Dr. John Doe,” prominently at the top, indicating the user’s profile. Below, key information is presented: the doctor’s date of birth (DOB), age (44), gender (Male), and job title (Ophthalmologist). This dashboard serves as a central hub for the doctor, allowing quick access to essential details. At the bottom, navigation options such as ”Home,” ”Scan,” and ”Results” suggest additional functionalities for managing patient information and related tasks, contributing to a streamlined user experience.

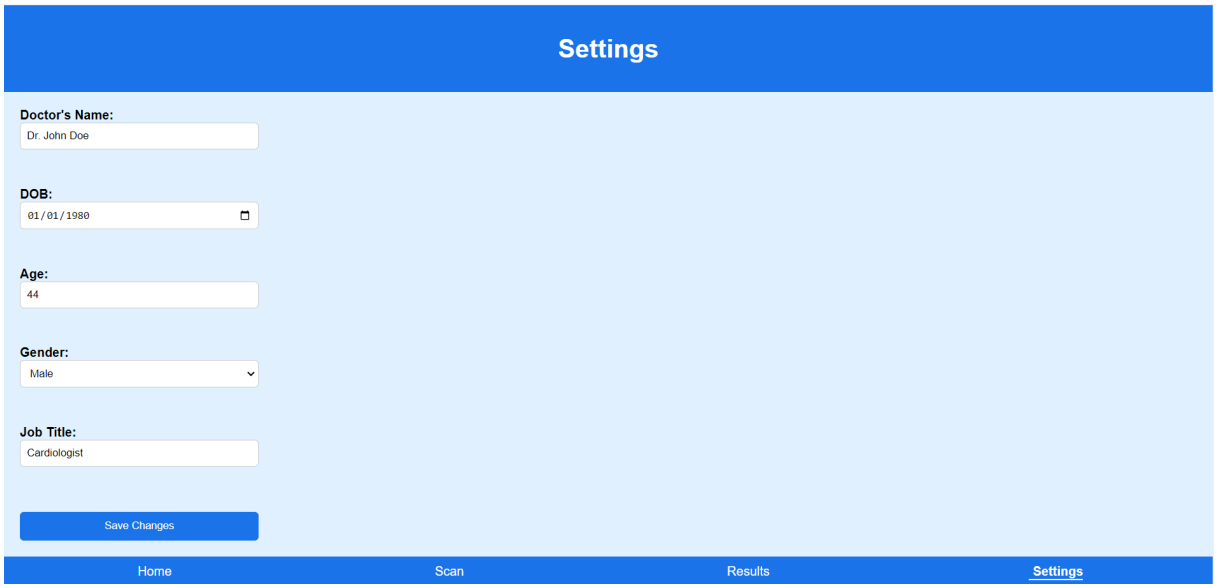


Figure 7.9: Eye Disease Classifier settings page .

Figure 7.9 displays a user interface for a ”Doctor’s Dashboard.” The layout includes the doctor’s name, ”Dr. John Doe,” prominently at the top, indicating the user’s profile. Below, key information is presented: the doctor’s date of birth (DOB), age (44), gender (Male), and job title (Ophthalmologist). This dashboard serves as a central hub for the doctor, allowing quick access to essential details. At the bottom, navigation options such as ”Home,” ”Scan,” and ”Results” suggest additional functionalities for managing patient information and related tasks, contributing to a streamlined user experience.

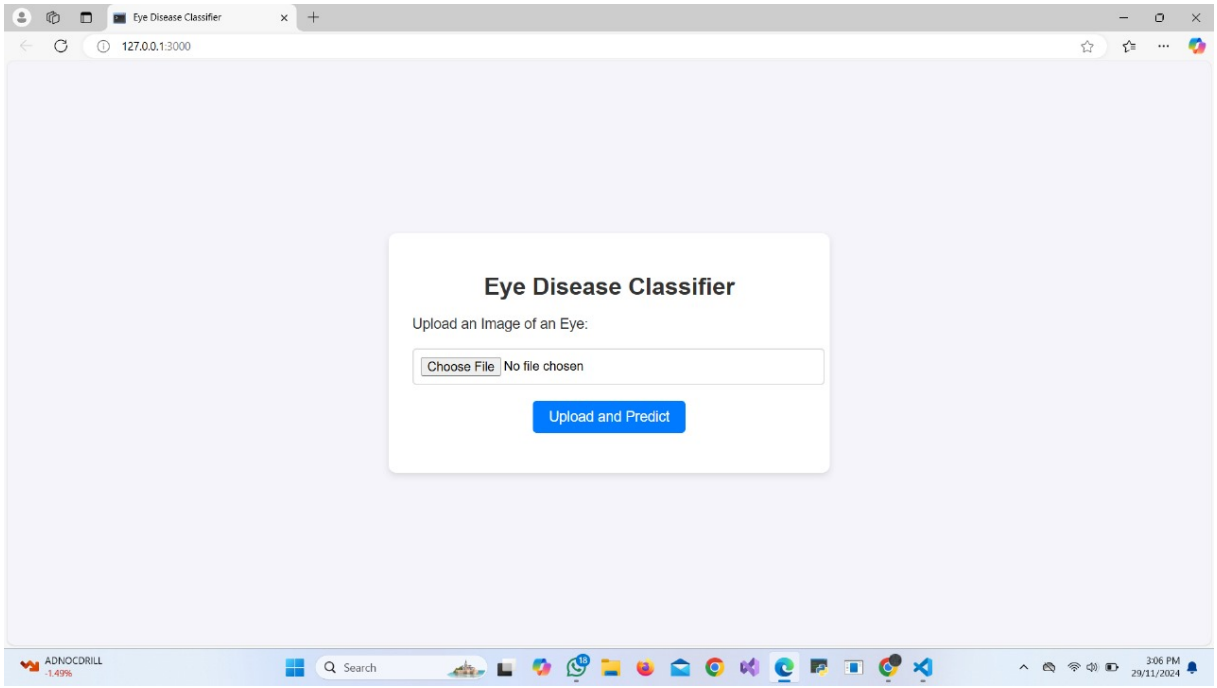


Figure 7.10: Eye Disease Classifier Web Page(1).

The ”Eye Disease Classifier” website serves as a valuable tool for ophthalmologists to upload images of patients’ eyes for disease analysis (Figure 7.10). Upon entering the site, they are greeted with a clear title indicating its purpose. In the central section, there’s a prompt inviting ophthalmologists to upload an image, accompanied by a ”Choose File” button that allows them to browse their device for a suitable image file. After selecting an image, they can click the ”Upload and Predict” button to submit it for analysis.

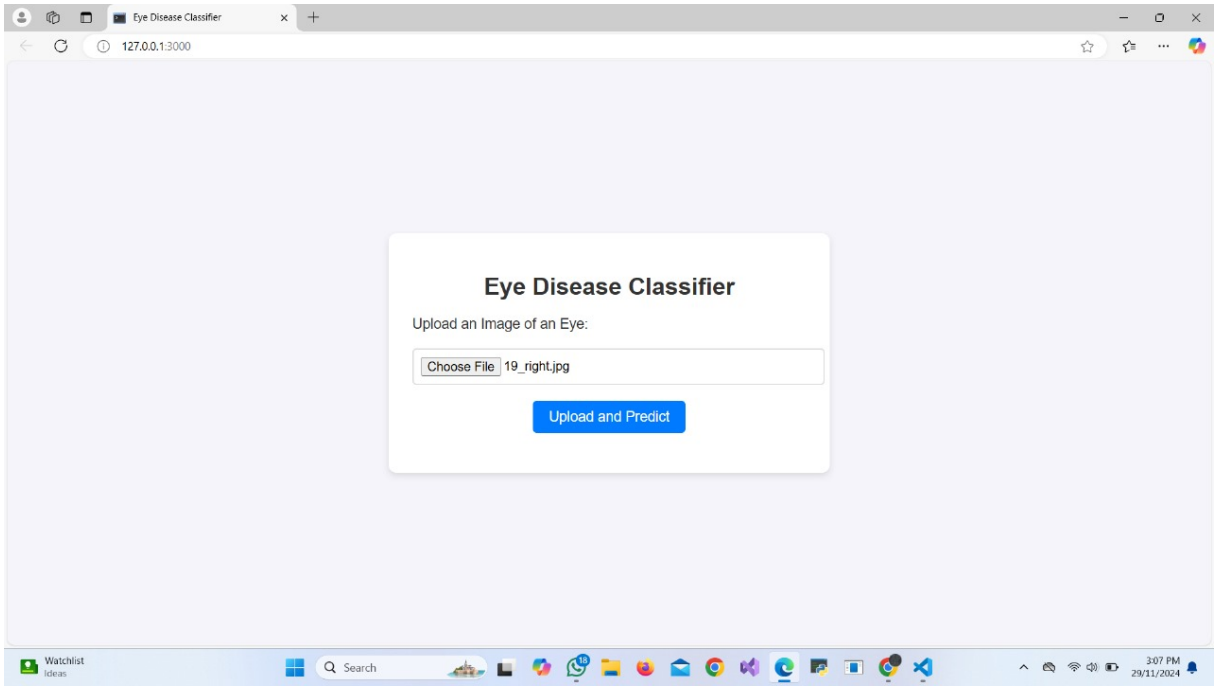


Figure 7.11: Eye Disease Classifier Web Page(2).

As it is showing in figure 7.11, in the center of the page, there is a prompt instructing eye doctors to ”Upload an image of an Eye.” A ”Choose File” button allows them to browse their device for an appropriate image file. Once an image is selected, such as ”right.jpg,” the doctor can click the ”Upload and Predict” button. This action submits the image for analysis, prompting the model to read and classify the uploaded image. The classification process involves analyzing the visual data to identify any potential eye diseases present in the image, such as diabetic retinopathy or glaucoma. This may take a moment depending on server processing.

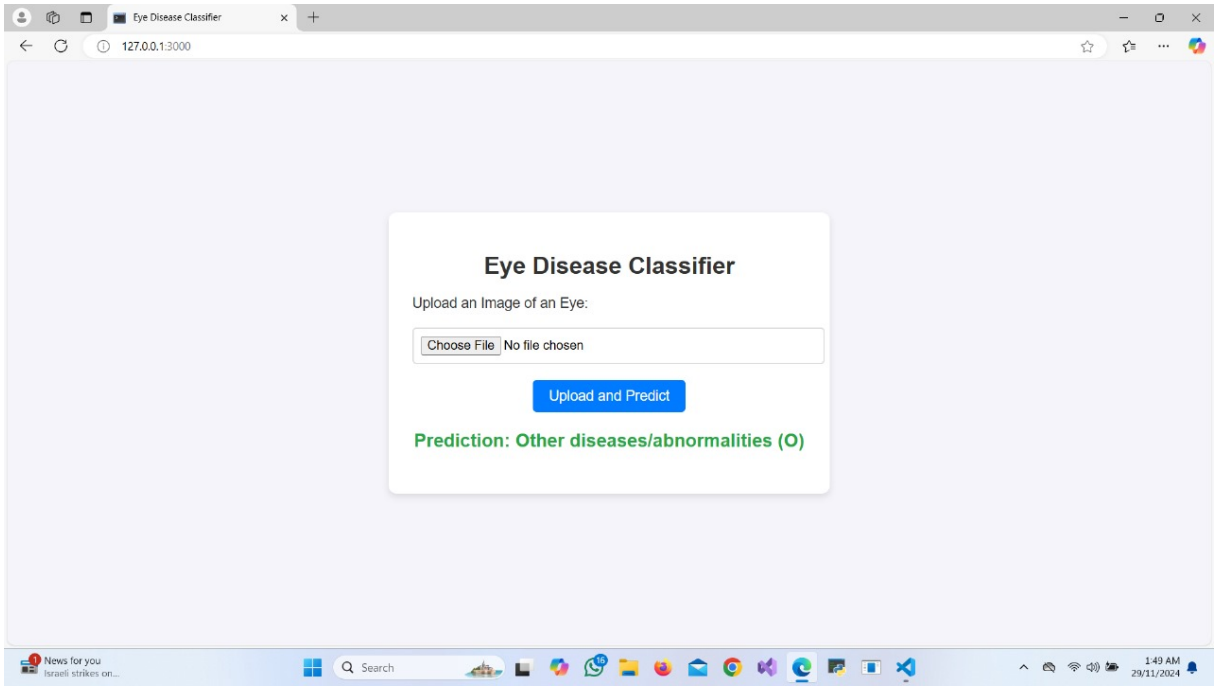


Figure 7.12: Eye Disease Classifier Web Page(3).

Once the user selects an image, they can click the ”Upload and Predict” button(figure 7.12), which submits the image for analysis. After the prediction process, users will see the results displayed below the button, indicating whether any diseases or abnormalities were detected, as shown by the message ”Prediction: Other diseases/abnormalities (0).” This section provides immediate feedback on the analysis, enhancing the user experience. To ensure accurate results, users should upload clear and well-lit images. The design encourages ease of use, making it accessible for individuals seeking to understand their eye health better.

7.5 Deep Learning model

The EyeScan model is built on a ResNet architecture, specifically chosen for its strength in feature extraction and handling complex image datasets like retinal scans. ResNet's residual blocks enable the model to learn intricate patterns and avoid vanishing gradients during training, making it particularly effective for identifying subtle abnormalities associated with various eye diseases. These residual connections allow the model to stack deeper layers without loss of information, improving its capacity to recognize detailed features.

Our model is designed for multi-class classification, allowing it to predict multiple eye conditions simultaneously. Each condition is represented as a unique class, and the model's output layer uses a softmax activation function to assign probabilities to each class. The disease with the highest probability is selected as the final prediction, ensuring that the output is interpretable and actionable.

To train the model, we employ categorical cross-entropy loss, which measures the divergence between the predicted probabilities and the true labels. This loss function ensures that the model improves its predictions for all classes, even when handling imbalanced datasets. For further robustness, we explore techniques like focal loss to give additional weight to underrepresented disease classes, enhancing the model's ability to detect rare conditions.

The model's outputs are refined through a fully connected layer that aggregates features extracted from ResNet. By passing through this layer, the model combines the detailed features into class-specific probabilities. For evaluation, we use metrics such as precision, recall, and F1 scores to ensure that the model performs well across all disease categories.

7.6 conclusion

This chapter detailed the comprehensive development of a mobile application and web platform for detecting eye diseases, utilizing a ResNet50 multi-class classification model. Our process began with extensive literature review, analyzing numerous studies and datasets to select one that aligned with our objectives.

We conducted experiments comparing various models, refining and improving their perfor-

mance to ensure accuracy in disease detection. The integration of advanced machine learning capabilities into both platforms aims to enhance diagnostic accuracy and user experience.

The Eye Scan app features a user-friendly interface designed for healthcare professionals, facilitating easy navigation and efficient image uploads for analysis. It provides immediate insights into potential eye conditions, thus promoting early diagnosis and patient education.

The accompanying web platform allows doctors to directly upload patient data, streamlining the diagnostic process with high accuracy. Both platforms reflect our commitment to improving accessibility and efficiency in ocular healthcare.

Looking ahead, we envision expanding the application's capabilities beyond image uploads to include direct eye capture functionality, allowing users to take real-time images of their retinas. Additionally, we aim to globalize the application, making it accessible to a wider audience and addressing diverse eye health needs worldwide.

Overall, the successful implementation of these tools demonstrates the potential of using pre-trained models for multi-class classification in real-time applications. This project not only showcases our findings but also sets the stage for future enhancements and broader applications in medical diagnostics, supported by the research we conducted throughout this endeavor.

Chapter 8

Conclusion

This project successfully demonstrated the potential of using machine learning and advanced image processing techniques for eye disease detection through a mobile application. Throughout the development process, we learned the importance of handling multi-class classification models, which allowed us to identify and categorize different eye conditions with high accuracy. Additionally, we gained valuable experience in developing both mobile applications and websites, honing skills in user interface design, backend integration, and deploying machine learning models in real-world applications.

Looking ahead, we plan to enhance the app's features by incorporating real-time image capture capabilities, eliminating the need for users to upload images manually. This upgrade will streamline the diagnostic process, making the app even more user-friendly and efficient. We also aim to improve the model's accuracy, expand the range of detectable conditions, and introduce more advanced features to further assist healthcare professionals and patients in early eye disease detection.

Appendix A

Source Code

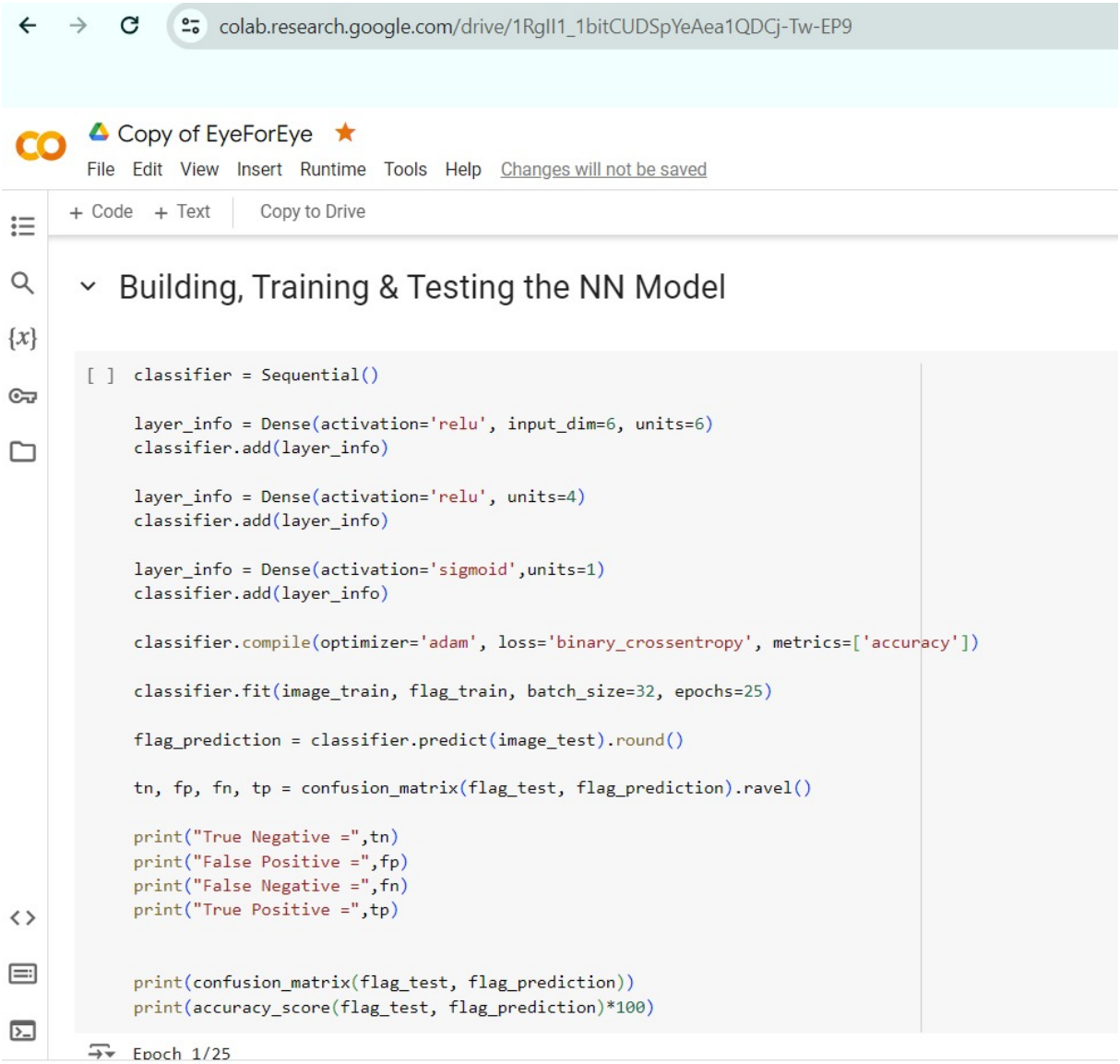


Figure A.1: Screenshot of the NN model

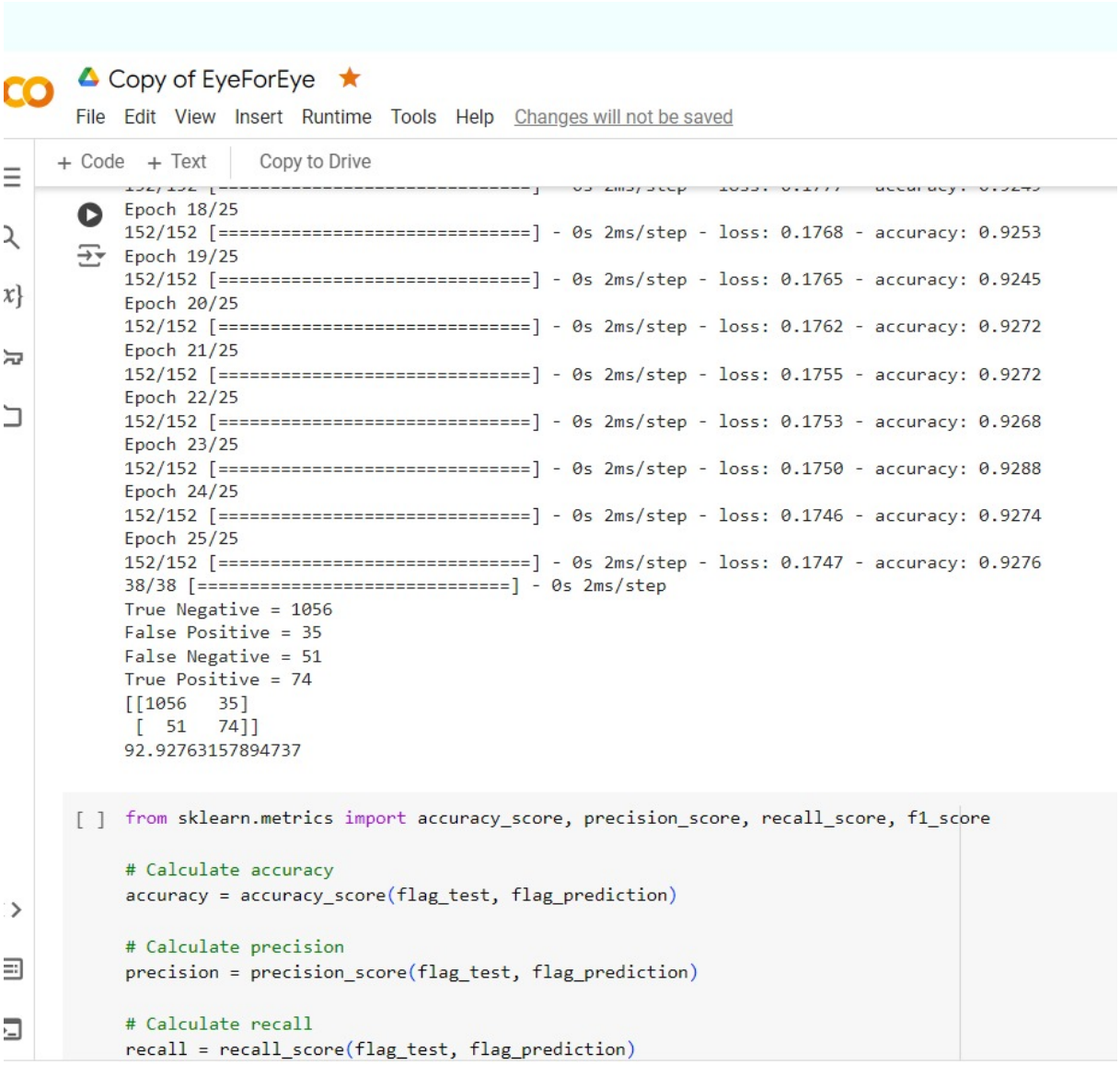


Figure A.2: Displaying the implementation of the code that shows the confusion matrix and the accuracy.

upCopy of Copy of Cataract Prediction using InceptionV3

File Edit View Insert Runtime Tools Help Last saved at May 12

+ Code + Text

```
[ ] model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 5, 5, 2048)        21802784

 flatten (Flatten)           (None, 51200)             0

 dense (Dense)               (None, 1)                 51201

=================================================================
Total params: 21853985 (83.37 MB)
Trainable params: 51201 (200.00 KB)
Non-trainable params: 21802784 (83.17 MB)
_________________________________________________________________

[ ] model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])

[ ] from tensorflow.keras.callbacks import ModelCheckpoint
    checkpoint = ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,save_best_only=True,
                                 save_weights_only=False,period=1)

WARNING:tensorflow:`period` argument is deprecated. Please use `save_freq` to specify the frequency in number of batches seen.

[ ] history = model.fit(x_train,y_train,batch_size=32,epochs=25,validation_data=(x_test,y_test),
                        verbose=1,callbacks=[checkpoint])

Epoch 1/25
28/28 [==============================] - ETA: 0s - loss: 21.1486 - accuracy: 0.7080WARNING:tensorflow:Can save best model only with val_acc available, skipping.
28/28 [==============================] - 17s 281ms/step - loss: 21.1486 - accuracy: 0.7080 - val_loss: 6.7474 - val_accuracy: 0.8578
Epoch 2/25
27/28 [===========================>..] - ETA: 0s - loss: 4.6569 - accuracy: 0.8646WARNING:tensorflow:Can save best model only with val_acc available, skipping.
28/28 [==============================] - 2s 86ms/step - loss: 4.6480 - accuracy: 0.8644 - val_loss: 4.4413 - val_accuracy: 0.9037
Epoch 3/25
27/28 [===========================>..] - ETA: 0s - loss: 1.7376 - accuracy: 0.9271WARNING:tensorflow:Can save best model only with val_acc available, skipping.
```

Figure A.3: Displaying the InceptionV3 model

upCopy of Copy of Cataract Prediction using InceptionV3

File Edit View Insert Runtime Tools Help Last saved at May 12

+ Code + Text

```
Accuracy: 0.9449541568756104
```

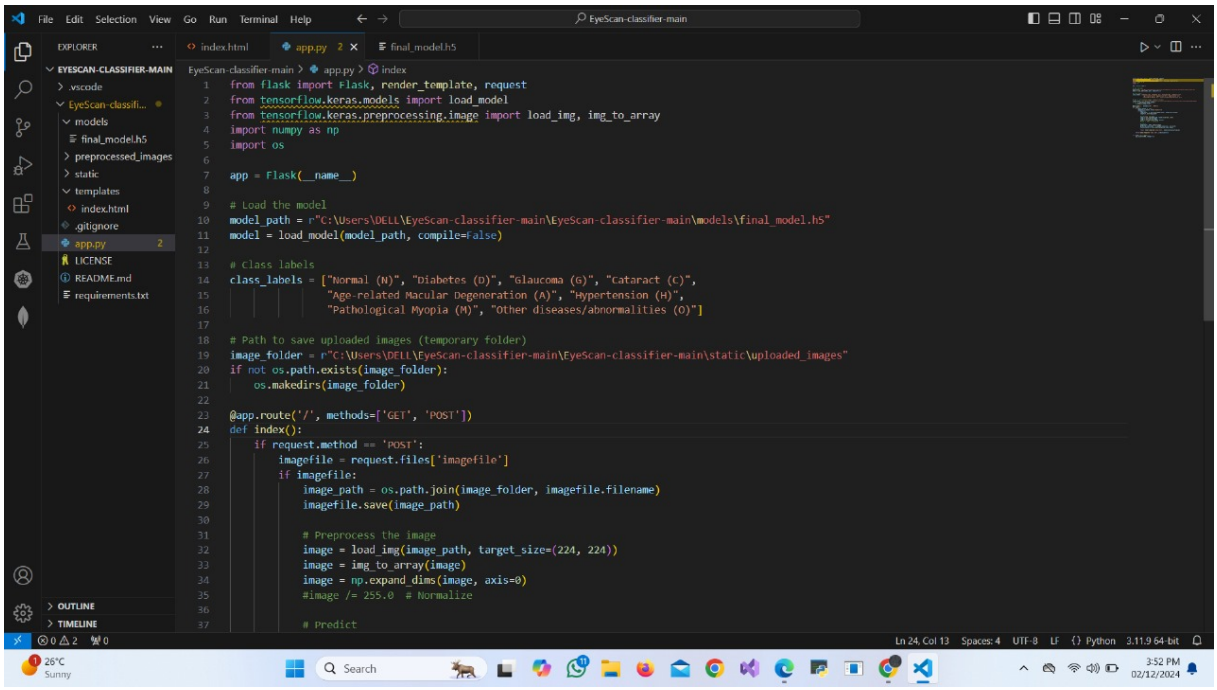
```
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score

# Assuming you have already defined your model and loaded the test data (x_test, y_test)

# Make predictions
y_pred = model.predict(x_test)
y_pred_binary = (y_pred > 0.5).astype('int32')

# Calculate confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred_binary)
print("Confusion Matrix:")
print(conf_matrix)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred_binary)
print("Accuracy:", accuracy)

# Calculate precision
precision = precision_score(y_test, y_pred_binary)
print("Precision:", precision)

# Calculate recall
recall = recall_score(y_test, y_pred_binary)
print("Recall:", recall)

# Calculate F1 score
f1 = f1_score(y_test, y_pred_binary)
print("F1 Score:", f1)
```

```
7/7 [==============================] - 4s 81ms/step
Confusion Matrix:
[[ 95   9]
 [  3 111]]
Accuracy: 0.944954128440367
Precision: 0.925
Recall: 0.9736842105263158
F1 Score: 0.9487179487179489
```

Figure A.4: Displaying the implementation of the code that shows the confusion matrix and the accuracy.

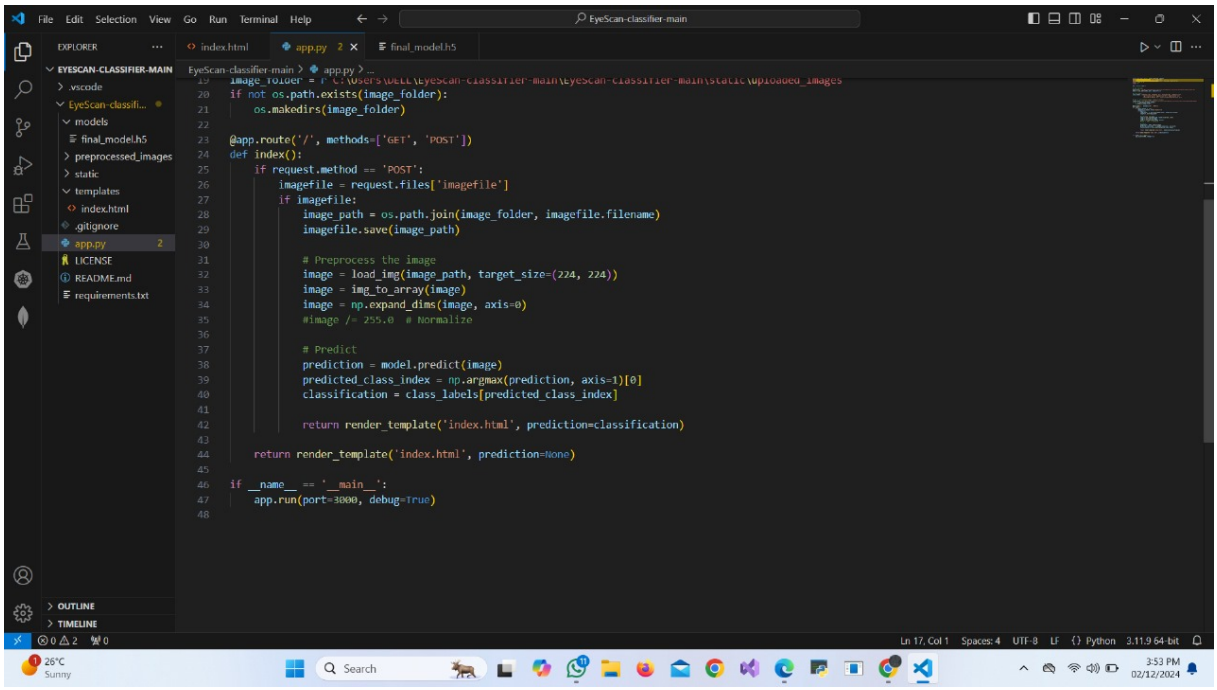
```python
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Example confusion matrix data
conf_matrix = [[211, 15],
               [22, 32]]

# Calculate evaluation metrics
accuracy = 0.9292763157894737
precision = 0.6788990825688074
recall = 0.592
f1_score = 0.6324786324786325

# Plot confusion matrix
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.heatmap(conf_matrix, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix')

# Bar chart for evaluation metrics
evaluation_metrics = ['Accuracy', 'Precision', 'Recall', 'F1 Score']
metrics_values = [accuracy, precision, recall, f1_score]

plt.subplot(1, 2, 2)
plt.bar(evaluation_metrics, metrics_values, color='skyblue')
plt.ylabel('Value')
plt.title('Evaluation Metrics')
plt.xlabel('Neural Network', labelpad=20)

plt.tight_layout()
plt.show()
```

Figure A.5: Displaying the implementation of the code that shows the confusion matrix and the accuracy.



```python
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import numpy as np
import os

app = Flask(__name__)

# Load the model
model_path = r"C:\Users\DELL\EyeScan-classifier-main\EyeScan-classifier-main\models\final_model.h5"
model = load_model(model_path, compile=False)

# Class labels
class_labels = ["Normal (N)", "Diabetes (D)", "Glaucoma (G)", "Cataract (C)",
                "Age-related Macular Degeneration (A)", "Hypertension (H)",
                "Pathological Myopia (M)", "Other diseases/abnormalities (O)"]

# Path to save uploaded images (temporary folder)
image_folder = r"C:\Users\DELL\EyeScan-classifier-main\EyeScan-classifier-main\static\uploaded_images"
if not os.path.exists(image_folder):
    os.makedirs(image_folder)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        imagefile = request.files['imagefile']
        if imagefile:
            image_path = os.path.join(image_folder, imagefile.filename)
            imagefile.save(image_path)

            # Preprocess the image
            image = load_img(image_path, target_size=(224, 224))
            image = img_to_array(image)
            image = np.expand_dims(image, axis=0)
            #image /= 255.0  # Normalize

            # Predict
```

Figure A.6: Screenshot of the ResNet Multi-Class classification code that is used to build our mobile application(1) .



```python
if not os.path.exists(image_folder):
    os.makedirs(image_folder)

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        imagefile = request.files['imagefile']
        if imagefile:
            image_path = os.path.join(image_folder, imagefile.filename)
            imagefile.save(image_path)

            # Preprocess the image
            image = load_img(image_path, target_size=(224, 224))
            image = img_to_array(image)
            image = np.expand_dims(image, axis=0)
            #image /= 255.0  # Normalize

            # Predict
            prediction = model.predict(image)
            predicted_class_index = np.argmax(prediction, axis=1)[0]
            classification = class_labels[predicted_class_index]

            return render_template('index.html', prediction=classification)

    return render_template('index.html', prediction=None)

if __name__ == '__main__':
    app.run(port=3000, debug=True)
```

Figure A.7: Screenshot of the ResNet Multi-Class classification code that is used to build our mobile application(2) .

References

[1] E. Hassan, M. S. Hossain, A. Saber, S. Elmougy, A. Ghoneim, and G. Muhammad, "A quantum convolutional network and resnet (50)-based classification architecture for the mnist medical dataset," *Biomedical Signal Processing and Control*, vol. 87, p. 105560, 2024.

[2] N. Li, T. Li, C. Hu, K. Wang, and H. Kang, "A benchmark of ocular disease intelligent recognition: One shot for multi-disease detection," in *Benchmarking, Measuring, and Optimizing: Third BenchCouncil International Symposium, Bench 2020, Virtual Event, November 15–16, 2020, Revised Selected Papers 3*. Springer, 2021, pp. 177–193.

[3] S. M. Sam, K. Kamardin, N. N. A. Sjarif, N. Mohamed *et al.*, "Offline signature verification using deep learning convolutional neural network (cnn) architectures googlenet inception-v1 and inception-v3," *Procedia Computer Science*, vol. 161, pp. 475–483, 2019.

[4] M. Bilgehan, "Comparison of anfis and nn models—with a study in critical buckling load estimation," *Applied Soft Computing*, vol. 11, no. 4, pp. 3779–3791, 2011.

[5] R. Bourne, J. D. Steinmetz, S. Flaxman, P. S. Briant, H. R. Taylor, S. Resnikoff, R. J. Casson, A. Abdoli, E. Abu-Gharbieh, A. Afshin *et al.*, "Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the global burden of disease study," *The Lancet global health*, vol. 9, no. 2, pp. e130–e143, 2021.

[6] J. Chua, M. Baskaran, P. G. Ong, Y. Zheng, T. Y. Wong, T. Aung, and C.-Y. Cheng, "Prevalence, risk factors, and visual features of undiagnosed glaucoma: the singapore

epidemiology of eye diseases study," *JAMA ophthalmology*, vol. 133, no. 8, pp. 938–946, 2015.

[7] Ş. AYKAT and S. Senan, "Using machine learning to detect different eye diseases from oct images," *International Journal of Computational and Experimental Science and Engineering*, vol. 9, no. 2, pp. 62–67, 2023.

[8] I. Keerthana and R. S. Kumar, "A survey of deep learning models to detect and classify eye disorders," in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE, 2023, pp. 30–36.

[9] M. Raju, K. P. Shanmugam, and C.-R. Shyu, "Application of machine learning predictive models for early detection of glaucoma using real world data," *Applied Sciences*, vol. 13, no. 4, p. 2445, 2023.

[10] K. Zhang, X. Liu, F. Liu, L. He, L. Zhang, Y. Yang, W. Li, S. Wang, L. Liu, Z. Liu *et al.*, "An interpretable and expandable deep learning diagnostic system for multiple ocular diseases: qualitative study," *Journal of medical Internet research*, vol. 20, no. 11, p. e11144, 2018.

[11] R. Sarki, "Automatic detection of diabetic eye disease through deep learning using fundus images," Ph.D. dissertation, Victoria University, 2021.

[12] J. D. Akkara and A. Kuriakose, "Role of artificial intelligence and machine learning in ophthalmology," *Kerala Journal of Ophthalmology*, vol. 31, no. 2, pp. 150–160, 2019.

[13] Q. Abbas, M. Albathan, A. Altameem, R. S. Almakki, and A. Hussain, "Deep-ocular: Improved transfer learning architecture using self-attention and dense layers for recognition of ocular diseases," *Diagnostics*, vol. 13, no. 20, p. 3165, 2023.

[14] R. Chavan and D. Pete, "Automatic multi-disease classification on retinal images using multilevel glowworm swarm convolutional neural network," *Journal of Engineering and Applied Science*, vol. 71, no. 1, p. 26, 2024.

[15] S. Mascarenhas and M. Agarwal, "A comparison between vgg16, vgg19 and resnet50 architecture frameworks for image classification," in *2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON)*, vol. 1. IEEE, 2021, pp. 96–99.

[16] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, M. Adam, A. Gertych, and R. San Tan, "A deep convolutional neural network model to classify heartbeats," *Computers in biology and medicine*, vol. 89, pp. 389–396, 2017.

[17] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: an overview," 2020. [Online]. Available: https://arxiv.org/abs/2008.05756

[18] A. Keswani, T. Jain, and B. Sharma, "Multi-class text classification using machine learning deep learning," in *2023 2nd International Conference on Futuristic Technologies (INCOFT)*, 2023, pp. 1–6.

[19] A. H.-L. Chan, R. H. Chan, and L. Dai, "Multi-classification using one-versus-one deep learning strategy with joint probability estimates," 2023. [Online]. Available: https://arxiv.org/abs/2306.09668