BATCH NO: MAD18

# EYESIGHT: INTEGRATIVE AI FOR NON-INVASIVE DIABETIC RETINOPATHY DETECTION USING PUPILLOMETRY AND ENSEMBLE DEEP LEARNING

*Major project report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**By**


| | | |
|---|---|---|
| **KANAGALA VINEELA** | (21UECS0261) | **(VTU 19936)** |
| **MALGIREDDY SIRI CHANDANA** | (21UECS0353) | **(VTU 19966)** |
| **J SHIVA SAI CHAITANYA** | (21UECM0099) | **(VTU 19928)** |

*Under the guidance of*
*Mrs.N.MOHANA SUGANTHI,M.E.,*
*ASSISTANT PROFESSOR*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2025**

**BATCH NO: MAD18**

# EYESIGHT: INTEGRATIVE AI FOR NON-INVASIVE DIABETIC RETINOPATHY DETECTION USING PUPILLOMETRY AND ENSEMBLE DEEP LEARNING

*Major project report submitted*
*in partial fulfillment of the requirement for award of the degree of*

**Bachelor of Technology**
**in**
**Computer Science & Engineering**

**By**


| | | |
|---|---|---|
| **KANAGALA VINEELA** | (21UECS0261) | **(VTU 19936)** |
| **MALGIREDDY SIRI CHANDANA** | (21UECS0353) | **(VTU 19966)** |
| **J SHIVA SAI CHAITANYA** | (21UECM0099) | **(VTU 19928)** |

*Under the guidance of*
*Mrs.N.MOHANA SUGANTHI,M.E.,*
*ASSISTANT PROFESSOR*

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**
**SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Deemed to be University Estd u/s 3 of UGC Act, 1956)**
**Accredited by NAAC with A++ Grade**
**CHENNAI 600 062, TAMILNADU, INDIA**


**May, 2025**

# CERTIFICATE

It is certified that the work contained in the project report titled " EYE SIGHT: INTEGRATIVE AI FOR NON-INVASIVE DIABETIC RETINOPATHY DETECTION USING PUPILLOMETRY AND ENSEMBLE DEEP LEARNING " by KANAGALA VINEELA (21UECS0261), MALGIREDDY SIRI CHANDANA (21UECS0353), J SHIVA SAI CHAITANYA (21UECM0099) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

**Signature of Supervisor**
**Mrs.N.Mohana Suganthi**
**Assistant Professor**
**Computer Science & Engineering**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science and Technology**

**Signature of Head/Assistant Head of the Department**
**Dr. N. Vijayaraj/Dr. M. S. Murali dhar**
**Professor & Head/ Professor &Assistant Head**
**Computer Science & Engineering**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science and Technology**

**Signature of the Dean**
**Dr. S P. Chokkalingam**
**Professor & Dean**
**School of Computing**
**Vel Tech Rangarajan Dr. Sagunthala R&D**
**Institute of Science and Technology**

# DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(KANAGALA VINEELA
Date: / /

(Signature)
(MALGIREDDY SIRI CHANDANA)
Date: / /

(Signature)
(J SHIVA SAI CHAITANYA)
Date: / /

# APPROVAL SHEET

This project report entitled EYE SIGHT: INTEGRATIVE AI FOR NON-INVASIVE DIABETIC RETINOPATHY DETECTION USING PUPILLOMETRY AND ENSEMBLE DEEP LEARNING by KANAGALA VINEELA (21UECS0261), MALGIREDDY SIRI CHANDANA (21UECS0353), J SHIVA SAI CHAITANYA (21UECM0099) is approved for the degree of B.Tech in Computer Science & Engineering.

**Examiners** **Supervisor**

Mrs.N.MOHANA SUGANTHI,M.E.,
Assistant Professor.

**Date: / /**
**Place:**

#### ACKNOWLEDGEMENT

We express our deepest gratitude to our **Honorable Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (Electrical), B.E. (Mechanical), M.S (Automobile), D.Sc., and Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.,Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We express our sincere thanks to our respected Chairperson and Managing Trustee **Mrs. RANGARAJAN MAHALAKSHMI KISHORE,B.E., Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, for her blessings.**

We are very much grateful to our beloved **Vice Chancellor Prof. Dr.RAJAT GUPTA,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean , School of Computing, Dr. S P. CHOKKALINGAM, M.Tech., Ph.D., & Professor & Associate Dean , School of Computing, Dr. V. DHILIP KUMAR,M.E.,Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Professor & Head, Department of Computer Science & Engineering, Dr. N. VIJAYARAJ, M.E., Ph.D.,** and **Professor & Assistant Head, Department of Computer Science & Engineering, Dr. M. S. MURALI DHAR, M.E., Ph.D.,**for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mrs.N.MOHANA SUGANTHI,M.E.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Dr. SADISH SENDIL MURUGARAJ,Professor, Dr.S.RAJAPRAKASH, M.E,Ph.D., Mr. V. ASHOK KUMAR, B.E,M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

**KANAGALA VINEELA (21UECS0261)**
**MALGIREDDY SIRI CHANDANA (21UECS0353)**
**J SHIVA SAI CHAITANYA (21UECM0099)**

## ABSTRACT

Diabetic Retinopathy (DR) remains one of the leading causes of vision impairment globally, particularly among individuals with long-standing diabetes. Traditional DR diagnostic techniques rely heavily on retinal imaging, which often involves expensive equipment, skilled interpretation, and invasive procedures. Project presents EyeSight, a novel, AI-powered system for the non-invasive detection of Diabetic Retinopathy using pupillometry-the study of pupil behavior in response to stimuli.The system captures pupillary response data and processes it using an ensemble of deep learning models, including ResNet, DenseNet, and EfficientNet. These models are trained to classify DR severity based on temporal variations in pupil dynamics, leveraging their individual strengths through a weighted ensemble strategy. And the system is deployed via a real-time, user-friendly Streamlit interface that enables clinicians to upload pupillometry data and receive immediate, interpretable results.Experimental evaluation demonstrates a high classification accuracy of 93.8%, along with robust precision, recall, and AUC scores, affirming the efficacy of ensemble learning in this novel application domain. The EyeSight system addresses the critical gaps in current DR screening technologies by offering a scalable, cost-effective, and non-invasive solution. It is especially suited for deployment in rural and resource-limited healthcare environments, where access to retinal imaging equipment is constrained.



**Keywords: Diabetic Retinopathy, Pupillometry, Deep Learning, Ensemble Models, Non-invasive Screening, ResNet, DenseNet, EfficientNet, Real-time Classification**

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AI | Artificial Intelligence |
| AUC | Area under the curve |
| CNN | Convolutional Neural Network |
| DR | Diabetic Retinopathy |
| ML | Machine Learning |
| OCT | optical coherence tomography |
| WHO | World Health Organization |
| XAI | Explainable Artificial Intelligence |

# TABLE OF CONTENTS

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

Diabetes mellitus is a chronic metabolic disorder that affects millions of individuals worldwide, with its incidence growing at an alarming rate. One of the most debilitating complications of diabetes is Diabetic Retinopathy (DR), a progressive ocular disease resulting from damage to the blood vessels in the retina. DR remains one of the leading causes of vision impairment and preventable blindness, particularly among individuals in their productive years. According to the World Health Organization (WHO), nearly 93 million people suffer from DR globally, and this number is expected to rise as diabetes becomes more prevalent.

The critical challenge in combating DR lies in its asymptomatic nature during the early stages. By the time symptoms appear, substantial and often irreversible retinal damage may have already occurred. Hence, early diagnosis and timely intervention are crucial to preventing vision loss. Unfortunately, traditional diagnostic methods such as fundus photography, optical coherence tomography (OCT), and fluorescein angiography are expensive, invasive, and require expert ophthalmologists to interpret the results accurately. These requirements limit their accessibility, especially in low-resource settings or rural regions where healthcare infrastructure is limited. In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as powerful tools in the field of medical diagnostics, offering potential solutions to many of these barriers. Deep Learning models, in particular, have demonstrated impressive capabilities in image classification and pattern recognition tasks, including retinal image analysis for DR detection. However, most existing AI-based DR diagnostic systems continue to rely on retinal fundus images, thereby inheriting the same limitations as traditional techniques.

To address these limitations, we propose ”EyeSight”, an integrative AI system designed for the non-invasive detection and classification of Diabetic Retinopathy using pupillometry. Pupillometry refers to the measurement and analysis of pupil size and reactivity under different lighting conditions or visual stimuli. Pupillary response is controlled by both the autonomic nervous system and the retinal circuitry; any damage to these pathways, such as that caused by diabetes, can result in abnormal pupil dynamics. These subtle deviations in pupillary behavior can be captured and analyzed using AI models to identify early signs of retinal dysfunction. Our system leverages ensemble deep learning architectures including ResNet, DenseNet, and EfficientNet to process and classify pupillometric data. The integration of multiple models enhances diagnostic accuracy, mitigates model bias, and ensures robustness across diverse input variations. The final output is delivered via a Streamlit-based user interface that enables clinicians or healthcare providers to upload pupillometric data and receive real-time predictions regarding the presence and severity of DR. Through this project, we aim to democratize DR screening by creating a reliable, non-invasive, and accessible diagnostic tool that has the potential to be deployed in both clinical and field settings. Our approach not only eliminates the need for expensive imaging equipment but also opens up new possibilities for early screening and monitoring of DR in underserved communities.

## 1.2 Background

Diabetic Retinopathy is a microvascular complication of diabetes, characterized by progressive damage to the retinal capillaries, leading to hemorrhages, microaneurysms, macular edema, and neovascularization. Clinically, DR progresses through distinct stages—from mild non-proliferative changes to proliferative retinopathy, which can cause retinal detachment and complete vision loss. The silent progression of DR and its irreversible consequences underscore the urgent need for early and frequent screening. Despite the importance of routine screening, several barriers hinder the widespread adoption of current DR detection methods:

- Cost: Fundus cameras and OCT machines are expensive and not widely available in low-resource settings.
- Invasiveness: Techniques like fluorescein angiography involve dye injection, which can cause discomfort and allergic reactions.
- Specialist Dependency: Diagnosis requires interpretation by trained ophthalmologists or retina specialists, which limits scalability.
- Accessibility: Rural populations often lack access to eye care centers and diagnostic infrastructure.

To overcome these limitations, researchers and clinicians have started exploring non-invasive biomarkers that can indicate early retinal dysfunction. One such promising biomarker is pupillary response, which reflects the function of retinal photoreceptors, optic nerves, and neural pathways. In diabetic patients, the autonomic control of the pupil can become impaired, leading to measurable changes in parameters such as:

- Baseline pupil size
- Latency of constriction and dilation
- Amplitude of response
- Recovery time post-stimulus

These parameters, when recorded and analyzed, can reveal subtle abnormalities indicative of underlying retinopathy. Advancements in sensor technology now allow for precise, contactless pupillometry using infrared cameras or wearable devices, making it feasible to collect such data in a variety of settings. The second major technological enabler for our project is deep learning, particularly Convolutional Neural Networks (CNNs) and ensemble learning techniques. CNNs are designed to extract hierarchical patterns and features from image-based data, which makes them suitable for analyzing pupil dynamics represented as time-series plots, videos, or segmented image frames. By combining the outputs of multiple CNN models (ResNet, DenseNet, EfficientNet), we can achieve higher classification accuracy and reduce model overfitting—thereby improving reliability.

The EyeSight project is built at the intersection of these two breakthroughs: non-invasive biometric sensing and deep learning-based classification. It envisions a future where DR screening can be performed rapidly, cost-effectively, and without discomfort—transforming preventive eye care across demographics.

### 1.3 Objective

The overarching goal of this project is to design, implement, and evaluate a non-invasive AI-based system for Diabetic Retinopathy detection using pupillometry and ensemble deep learning. The specific objectives are as follows:

1. Data Acquisition: To develop a reliable data collection pipeline using pupillometry techniques, capturing pupil response under controlled stimulus conditions.

2. Preprocessing and Feature Engineering: To clean, normalize, and augment raw pupillometric data for deep learning. This includes noise reduction, feature extraction, and data labeling.
3. Model Development:
   - To implement and train three advanced CNN models: ResNet, DenseNet, and EfficientNet for DR classification.
   - To apply ensemble learning strategies such as majority voting or weighted averaging to combine model predictions and improve diagnostic performance.
4. System Integration:
   - To create an intuitive Streamlit-based web interface for clinicians to upload data and view DR classifications.
   - To ensure real-time response capability and interactive output interpretation.
5. Validation and Testing:
   - To rigorously test the system using both synthetic and real-world datasets.
   - To evaluate the model performance using key metrics: accuracy, precision, recall, F1-score, and AUC (Area Under Curve).
6. Deployment and Real-World Evaluation:
   - To deploy the trained ensemble model in a cloud or local clinical setting.
   - To test the usability and effectiveness of the system in real-time healthcare environments.
7. Enhancements: To explore personalization, historical data tracking, and mobile interface development in future phases.

## 1.4 Problem Statement

The early detection of diabetic retinopathy is critical to prevent irreversible vision loss. However, existing diagnostic techniques rely heavily on invasive and costly retinal imaging, which limits their scalability and accessibility—particularly in underserved or remote populations. Furthermore, the dependence on expert ophthalmologists and high-resolution imaging devices restricts mass screening, and the discomfort associated with invasive diagnostic procedures often deters patient compliance. Additionally, while AI-based tools have started entering the clinical landscape, most still operate on image-based data, posing challenges such as data privacy, model generalization, and infrastructure requirements.

Given these challenges, the core problem addressed in this project can be formulated as: ”How can we develop a non-invasive, AI-powered, real-time system for detecting and classifying diabetic retinopathy by analyzing pupillary response data using ensemble deep learning models, thereby enhancing accessibility, affordability, and diagnostic accuracy in real-world healthcare settings” The EyeSight system aims to bridge this gap by introducing a new, contactless diagnostic modality backed by deep learning. It leverages the biological relevance of pupillary changes in diabetic individuals and the computational power of ensemble CNNs to provide a fast, reliable, and user-friendly screening solution that can function in both clinical and community-level deployments.

# Chapter 2

# LITERATURE REVIEW

[1] Ai, Z et.al.,(2021) a deep ensemble learning algorithm that integrates attention mechanisms for improved detection of diabetic retinopathy. The method showcases the potential of combining deep learning with attention-based models to increase detection accuracy. While their algorithm offers promising results, it lacks the integration of non-invasive data, which the proposed system will incorporate through pupillometry and ensemble deep learning models.

[2] Bidwai et.al.,(2022) conducted a systematic review on AI approaches for diabetic retinopathy detection. The authors provide an extensive analysis of various machine learning models, including CNN-based and ensemble learning techniques, which have been applied to retinal image analysis. While their review highlights the effectiveness of AI in DR detection, it predominantly focuses on image-based diagnostics, which may not be as accessible in resource-limited settings compared to the non-invasive pupillometry approach being proposed in Eyesight.

[3] Farooq, M et.al.,(2022) discussed the development of a computer-aided diagnostic system for screening diabetic retinopathy based on deep learning techniques. Their system highlights advancements in AI for DR detection, but their work primarily uses image data. The proposed system's unique approach will integrate pupillometry as a non-invasive diagnostic tool, offering a novel angle in DR detection.

[4] Huang, X et.al.,(2022)introduced an innovative approach in using artificial intelligence to promote the diagnosis and screening of diabetic retinopathy (DR). Their work emphasizes the application of AI models to enhance the diagnostic accuracy of DR, demonstrating significant improvements in efficiency compared to traditional methods. However, their system primarily focuses on imaging techniques, whereas the proposed system aims to introduce non-invasive pupillometry to provide a more patient-friendly alternative.

[5] Kumari, P. M et.al.,(2023) presented research on the automatic detection of genetic eye diseases in children using pupillometry. Study demonstrates the versatility

of pupillometry in identifying different eye diseases, supporting its potential integration into the Eyesight system. While the focus of their work is on genetic disorders, the proposed system will extend the use of pupillometry to diabetic retinopathy diagnosis.

[6] Munjral et.al.,(2022) reviewed the use of artificial intelligence in cardiovascular risk stratification in diabetic retinopathy, exploring its role in diagnosing DR and its relation to atherosclerotic pathways. Study highlights the potential of AI in multi-disease diagnosis, but it does not address the non-invasive nature of detection, which is a primary advantage of the proposed system.

[7] Rom, Y et.al.,(2022) developed a deep learning model that predicts the progression of diabetic retinopathy by analyzing non-invasive retinal imaging. It demonstrates the power of AI in understanding disease progression and aiding early intervention. However, their method still relies on imaging, unlike the proposed Eyesight system, which focuses on pupillometry as a less invasive alternative.

[8] Senapati, et.al.,(2024) conducted a systematic review that critically evaluates advancements in artificial intelligence (AI) for diabetic retinopathy detection, highlighting various AI methodologies, their diagnostic accuracy, and the potential for improving screening processes. The review underscores the transformative impact of AI in enhancing early detection and management of diabetic retinopathy, addressing both clinical implications and future research directions.

[9] Vujosevic et.al.,(2022) explored digital innovations for retinal care in diabetic retinopathy. Their work delves into cutting-edge digital technologies, such as AI-based systems and telemedicine, to improve DR management. Although their findings support the integration of technology into healthcare, they primarily discuss imaging-based innovations, while the proposed system aims to enhance accessibility by utilizing non-invasive pupillometry and deep learning.

[10] Zhang, Z et.al.,(2023) proposed a non-invasive and affordable screening method for type 2 diabetes using deep learning-based risk assessment and ophthalmic images inspired by traditional Chinese medicine. Their approach emphasizes affordable and non-invasive methods, which aligns with the goals of the proposed system. However, their focus remains on ophthalmic imaging, while the Eyesight system will focus on pupillometry as an alternative non-invasive method for early DR detection.

## 2.1 Existing System

The diagnosis of Diabetic Retinopathy (DR) has traditionally relied on retinal imaging technologies such as fundus photography, fluorescein angiography, and optical coherence tomography (OCT). These imaging techniques have enabled ophthalmologists to detect and monitor key symptoms of DR including microaneurysms, hemorrhages, hard exudates, and neovascularization. Fundus photography captures the retinal surface for manual inspection, OCT offers cross-sectional images of the retina, and fluorescein angiography uses dye to highlight retinal blood vessels. Together, these methods represent the clinical gold standard for DR screening.

However, despite their diagnostic accuracy, existing systems are hindered by significant limitations. First and foremost is their dependency on expensive equipment and specialized personnel. The acquisition of high-resolution retinal images requires professional-grade cameras and imaging hardware, making such systems unaffordable for small clinics or rural healthcare centers. Additionally, interpretation of the results depends heavily on trained ophthalmologists or retinal specialists, contributing to diagnostic delays and bottlenecks in regions with a shortage of skilled professionals. Another concern is patient compliance and comfort. Techniques such as fluorescein angiography can be invasive and may cause discomfort, allergic reactions, or other side effects due to the injected dye. This reduces the appeal of frequent screening, especially among elderly or pediatric populations. Moreover, environmental factors such as lighting conditions and pupil dilation can influence image quality, adding variability to diagnosis. Although computer-aided diagnosis (CAD) systems have attempted to automate image analysis through traditional algorithms, their performance was often constrained by hardcoded features, noise sensitivity, and poor generalization. With the rise of deep learning, more sophisticated systems have emerged that leverage convolutional neural networks (CNNs) to automatically extract features and classify DR severity based on large datasets of retinal fundus images.

These models offer promising accuracy but still depend on retinal imaging and inherit the associated costs and accessibility barriers. Consequently, the need for a more accessible, non-invasive, and scalable alternative has prompted researchers to explore new modalities beyond retinal photography. Pupillometry, which studies the dynamic responses of the pupil to light and visual stimuli, is one such modality. Because the pupil's behavior is regulated by the autonomic nervous system and retinal photoreceptors, abnormal pupillary responses can indirectly signal the presence of DR. The EyeSight system builds upon this insight by combining pupillometry with ensemble deep learning models to enable non-invasive DR detection.

### 2.2 Related Work

The integration of artificial intelligence (AI) in the detection and diagnosis of diabetic retinopathy (DR) has seen significant advancements in recent years. For instance, Ai et al. (2021) developed a deep ensemble learning algorithm that incorporates attention mechanisms to enhance detection accuracy. While their approach demonstrates the potential of combining deep learning with attention-based models, it primarily relies on imaging data and does not consider non-invasive methods. In contrast, the proposed Eyesight system aims to incorporate pupillometry, a non-invasive diagnostic tool, to improve accessibility and patient comfort in DR detection. Similarly, Bidwai et al. (2022) conducted a systematic review of AI approaches for DR detection, highlighting various machine learning models, including CNN-based techniques. However, their focus on image-based diagnostics may limit accessibility in resource-limited settings, further emphasizing the need for non-invasive alternatives like pupillometry.

Moreover, the work of Huang et al. (2022) showcases the application of AI models to enhance diagnostic accuracy in DR, yet it predominantly emphasizes imaging techniques. This limitation is addressed by the proposed system, which seeks to introduce non-invasive pupillometry as a patient-friendly alternative. Additionally, Kumari et al. (2023) demonstrated the versatility of pupillometry in detecting genetic eye diseases, supporting its potential integration into the Eyesight system for DR diagnosis. The exploration of digital innovations by Vujosevic et al. (2022) further underscores the importance of integrating technology into healthcare, although their focus remains on imaging-based solutions. The proposed Eyesight system aims to bridge this gap by utilizing non-invasive pupillometry alongside deep learning models, thereby enhancing accessibility and early detection of diabetic retinopathy.

### 2.3 Research Gap

Although the field of diabetic retinopathy detection has witnessed significant advances due to deep learning and ensemble modeling techniques, several critical research gaps remain. The majority of state-of-the-art models rely almost exclusively on high-quality retinal images, which inherently limits their applicability in remote or resource-constrained environments. Despite the proven effectiveness of fundus-based diagnosis, such systems fail to address the global need for low-cost, non-invasive screening tools.

While some initiatives have successfully utilized pupillometry to diagnose neurological or congenital conditions, its application in diabetic retinopathy remains largely uncharted. This presents an untapped opportunity to develop systems that harness the autonomic nervous system's response to identify retinal dysfunction. Furthermore, ensemble learning approaches have typically been applied to image-based DR detection without considering how similar techniques can be adapted for other modalities like pupillometry.

The benefits of ensemble models-robustness, higher accuracy, and better generalization-have not been fully explored in the context of non-invasive pupil-based data. Moreover, there is a scarcity of research on integrating deep learning models into real-time web-based platforms that are accessible, scalable, and tailored for clinical workflows. Interpretability also remains a pressing issue.

Although attention mechanisms have improved the transparency of model predictions, clinicians continue to demand diagnostic tools that are explainable, trustworthy, and verifiable. This concern becomes more pronounced in the context of non-image data, where feature attribution and model behavior are more difficult to visualize and justify. Therefore, the EyeSight project addresses a unique confluence of these gaps: it applies ensemble deep learning techniques to a non-invasive modality (pupillometry), develops a user-centric real-time interface for clinicians, and opens a new direction in diabetic retinopathy diagnostics that is cost-effective, scalable, and biologically grounded. rephrase words its not research its a major project

# Chapter 3

# PROJECT DESCRIPTION

## 3.1 Existing System

Traditional systems for diabetic retinopathy (DR) detection predominantly revolve around image-based diagnostics using tools such as fundus cameras, fluorescein angiography, and optical coherence tomography (OCT). These methods enable clinicians to visually assess the retina for microvascular abnormalities that signify different stages of DR. While these techniques have been the gold standard in ophthalmic diagnostics, they present several significant limitations in practical deployment, especially in rural and resource-constrained environments.

Firstly, the requirement for expensive and specialized equipment makes the deployment of such systems infeasible in many parts of the world. The cost of a single fundus camera can exceed several thousand dollars, and these devices are often bulky, necessitating a clinical setting for operation. Secondly, these systems rely on trained ophthalmologists for image interpretation. Given the global shortage of eye care specialists, especially in low- and middle-income countries, large-scale screening is logistically impractical. Furthermore, image-based systems often require dilation of the pupils for optimal imaging, a process that can be uncomfortable and time-consuming for patients. Additionally, the process of image acquisition is sensitive to lighting conditions, patient cooperation, and motion artifacts. In real-world scenarios, this leads to inconsistent data quality, affecting the reliability of automated or manual interpretation. Even with advances in AI-driven image analysis, these systems are not fully autonomous, and their reliance on retinal imaging persists as a bottleneck.

In summary, while existing systems are effective in detecting DR, they are constrained by factors such as cost, invasiveness, need for expertise, and lack of portability. These limitations hinder early screening and delay timely intervention, particularly in vulnerable populations. Hence, there is a pressing need for an innovative system that overcomes these challenges.

### 3.2 Proposed System

The EyeSight project introduces a paradigm shift in diabetic retinopathy diagnosis by employing a non-invasive, AI-powered approach based on pupillometry. Instead of relying on traditional retinal imaging, the EyeSight system analyzes pupil responses to light and visual stimuli using data collected from infrared-based pupillometers or video sequences. The project leverages ensemble deep learning models, including ResNet, DenseNet, and EfficientNet, to detect subtle abnormalities in pupil dynamics that are indicative of retinal nerve or vascular damage due to diabetes. The core novelty of the proposed system lies in its integration of three fundamental advancements:

1. The use of pupillometric data as a non-invasive biomarker for DR,
2. The application of ensemble learning to improve diagnostic accuracy and robustness, and
3. The deployment of a Streamlit-based web application that enables real-time DR detection and feedback to clinicians.

Pupillometry allows the system to function without the need for high-resolution fundus images, thus reducing costs, increasing portability, and improving patient comfort. The ensemble learning strategy combines the predictions of multiple deep neural networks to reduce the chances of false positives and false negatives. Moreover, the proposed system includes comprehensive preprocessing of raw data—such as noise removal, normalization, and augmentation—to ensure high-quality input to the AI models. The EyeSight platform is designed to operate in real-time and is deployable in both clinical and non-clinical environments. Its interface allows healthcare professionals to upload pupil data and receive immediate analysis with accompanying severity levels. This democratizes access to early diagnosis and is particularly valuable in community health initiatives, mobile screening camps, and rural outreach programs.

### 3.3 Feasibility Study

The EyeSight project evaluates its economic, technical, and social aspects to determine its viability in enhancing diabetic retinopathy diagnosis. By utilizing a non-invasive, AI-powered approach based on pupillometry, EyeSight aims to improve accessibility and accuracy in detecting diabetic retinopathy, particularly in underserved populations. This comprehensive assessment will explore the project's potential for successful implementation, sustainability, and its overall impact on healthcare delivery and patient outcomes.

#### 3.3.1 Economic Feasibility

The EyeSight project is promising, as it presents a cost-effective alternative to traditional retinal imaging methods. Traditional screenings often require expensive equipment and specialized personnel, which can be a barrier in low-resource settings. EyeSight's use of infrared-based pupillometers and video sequences reduces costs significantly, making it more accessible. The real-time analysis provided by the Streamlit-based web application can lead to quicker diagnoses, potentially lowering long-term costs associated with untreated diabetic retinopathy. By improving early detection rates, EyeSight can alleviate the financial burden on healthcare systems by preventing advanced complications that necessitate costly interventions. Overall, the economic feasibility supports the potential for widespread adoption and integration into existing healthcare frameworks.

#### 3.3.2 Technical Feasibility

The EyeSight system is strong, supported by advanced ensemble deep learning models such as ResNet, DenseNet, and EfficientNet. These models are well-established in computer vision and have shown high accuracy in diagnostic applications. The innovative use of pupillometric data as a non-invasive biomarker for diabetic retinopathy leverages existing technology effectively. The system incorporates comprehensive preprocessing techniques, ensuring high-quality input for the AI models. Additionally, the user-friendly Streamlit-based web application allows healthcare professionals to upload data and receive immediate analysis, enhancing usability. The necessary technical infrastructure, including cloud computing for data processing and storage, is readily available and scalable, making EyeSight adaptable to various clinical and non-clinical environments.

#### 3.3.3 Social Feasibility

The EyeSight project is significant, as it addresses healthcare disparities in diabetic retinopathy diagnosis and management. By offering a non-invasive, cost-effective, and portable solution, EyeSight enhances access to early detection, particularly in rural and underserved communities. The system's real-time feedback empowers healthcare professionals to make informed decisions quickly, improving patient outcomes and fostering trust in the healthcare system. The user-friendly interface encourages adoption among clinicians who may be less familiar with advanced imaging technologies. Furthermore, EyeSight aligns with public health initiatives aimed at reducing diabetes-related complications and can be integrated into community health programs and mobile screening camps, promoting awareness and education about diabetic retinopathy. Overall, the social feasibility of EyeSight is robust, enhancing diagnostic capabilities and contributing to broader public health goals.

## 3.4 System Specification

The EyeSight system is composed of multiple hardware and software components. The hardware includes a pupillometry sensor or infrared camera capable of recording pupil reactions to light. The software includes modules for data acquisition, preprocessing, model inference, and result visualization. The backend is implemented in Python and includes deep learning models built using PyTorch. These models are trained on labeled pupillometric datasets, which may include both synthetic and real patient data.

The preprocessing unit applies techniques such as normalization, smoothing, and noise filtering. The inference engine hosts the ensemble models and generates classification outputs. The frontend is built using Streamlit, providing a browser-based interface for clinicians. The system requires access to computational resources such as a local GPU or cloud-hosted CPU/GPU instances for efficient model execution. The modular design ensures that the system can be adapted to future upgrades, including additional models, new datasets, and feature enhancements.

### 3.4.1 Tools and Technologies Used

The EyeSight system leverages a modern technology stack to deliver high performance and accessibility. Deep learning models are developed using PyTorch due to its flexibility and dynamic computation graph support. Data preprocessing is performed using libraries like NumPy, Pandas, and OpenCV. For data visualization and result plotting, Matplotlib and Seaborn are used. Model training workflows are enhanced with TensorBoard for monitoring metrics and performance across epochs. Ensemble strategies are implemented using Scikit-learn's voting classifiers, and advanced attention mechanisms are explored using PyTorch's Transformer utilities. The web interface is developed using Streamlit, which enables rapid prototyping and deployment of interactive data science applications. Backend logic is hosted on Flask when required, and cloud services such as Google Colab, AWS EC2, or Google Cloud Run are used for training and inference hosting. Version control is managed via Git, and collaborative development is maintained using GitHub.

#### 3.4.2 Standards and Policies

The EyeSight project adheres to key software and healthcare development standards to ensure safety, interoperability, and reproducibility. From a software engineering perspective, the system follows the IEEE 830 standard for Software Requirements Specification (SRS) and adheres to modular programming principles for maintainability.

From a data privacy standpoint, the system complies with HIPAA and GDPR guidelines concerning the handling of medical data, particularly sensitive biometric information like pupil videos. User consent, data encryption, and access controls are implemented to safeguard patient information.

In terms of model evaluation, standard performance metrics such as accuracy, precision, recall, F1-score, and AUC are used. Cross-validation techniques and stratified sampling are applied to ensure fairness and minimize bias. Documentation is maintained as per ISO/IEC standards to ensure replicability and clarity in development and deployment workflows.

Moreover, accessibility guidelines (WCAG 2.1) are considered while designing the user interface to ensure usability for users with visual impairments or other disabilities. This inclusive approach enhances the real-world applicability and ethical integrity of the EyeSight system.

# Chapter 4

# SYSTEM DESIGN AND METHODOLOGY

## 4.1 System Architecture

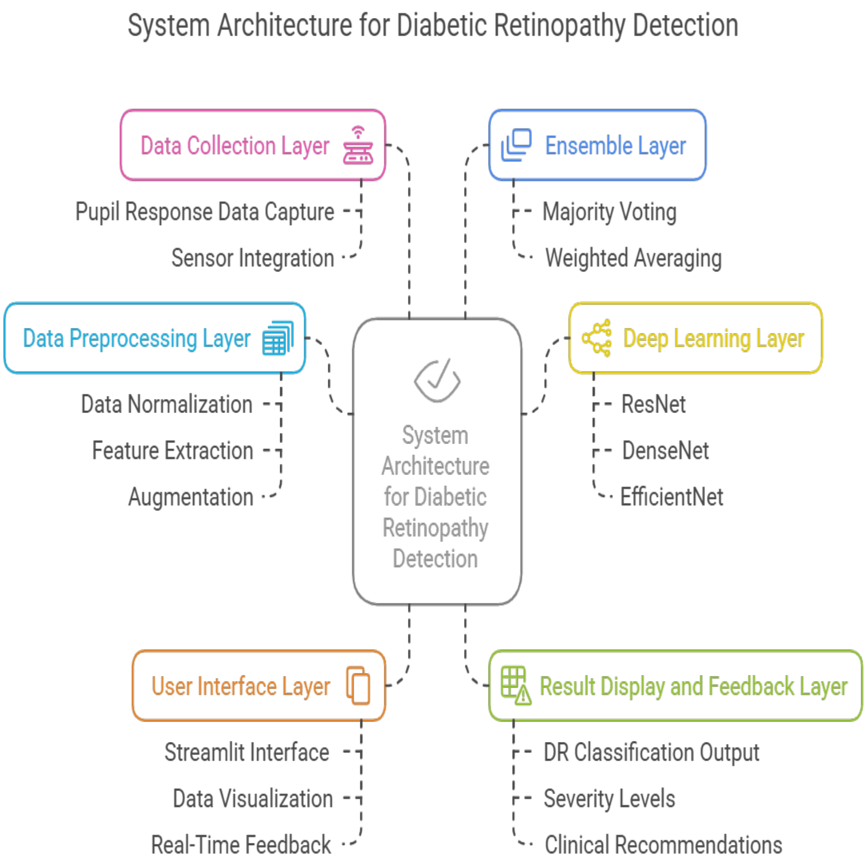


Figure 4.1: System Architecture

Figure 4.1 illustrates the modular architecture of the EyeSight system, comprising layers for data acquisition, preprocessing, model inference with ensemble deep learning, and a user-friendly presentation interface for clinicians. This design promotes efficient pupillometric data analysis, real-time visualization, and future adaptability.

## 4.2 Design Phase

### 4.2.1 Data Flow Diagram

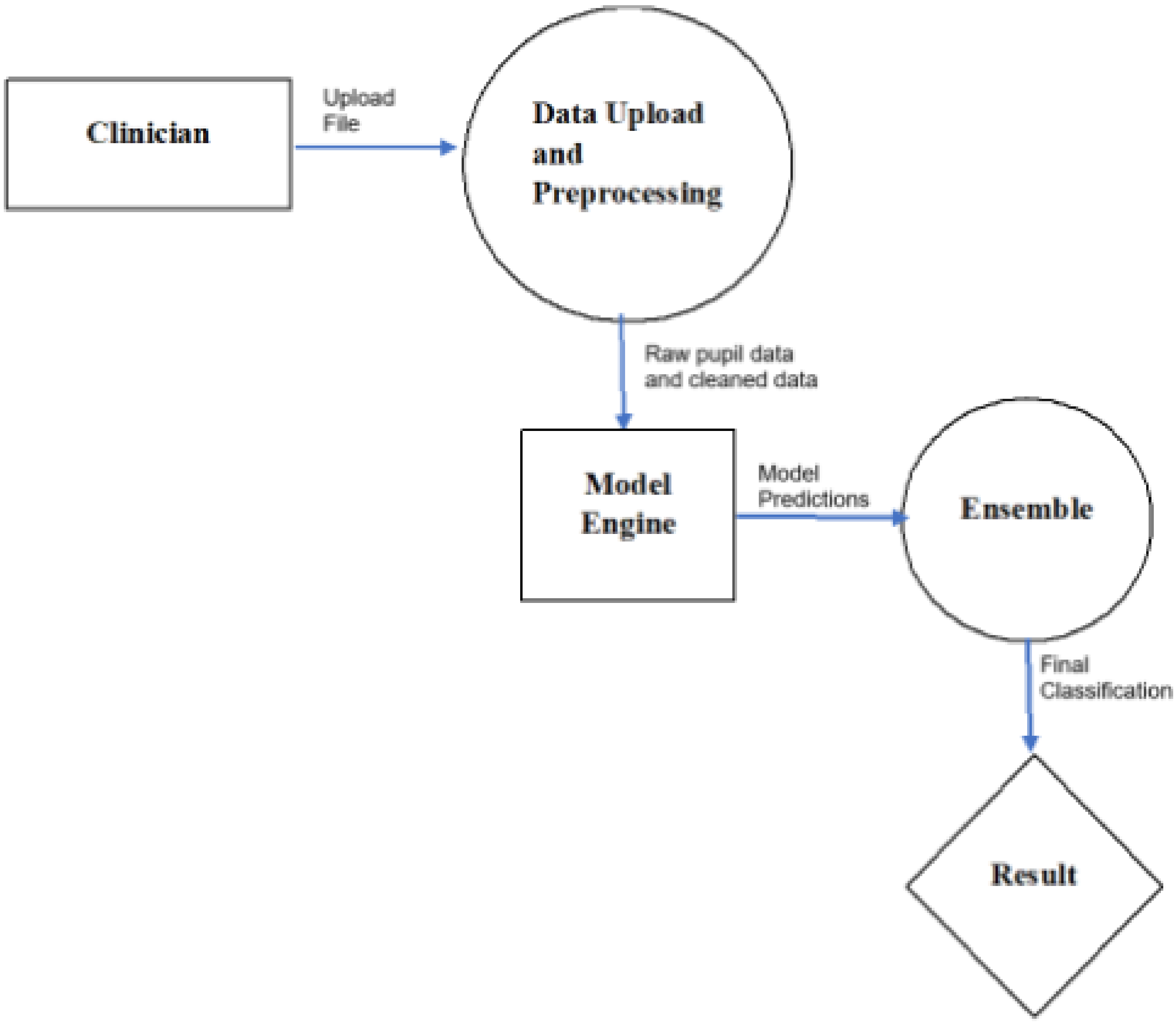


Figure 4.2: Data Flow Diagram

Figure 4.2 outlines the workflow of the EyeSight system, starting with the clinician uploading raw pupil data, which is then preprocessed for analysis. The cleaned data is processed by the Model Engine for predictions, aggregated by the Ensemble Aggregator for final classification, and presented to the clinician for insights.

### 4.2.2 Use Case Diagram

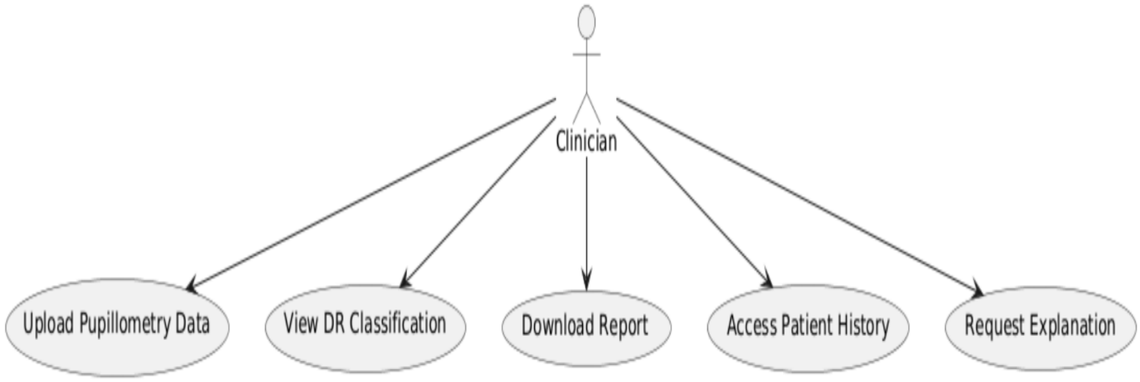


Figure 4.3: Use Case Diagram

Figure 4.3 illustrates a clinician's interactions with a digital system for managing patient eye health, featuring functions like uploading pupillometry data, viewing diabetic retinopathy classifications, downloading reports, accessing patient history, and requesting explanations. This interconnected approach enhances monitoring and management of patient care through efficient data assessments and informed decision-making.

### 4.2.3 Class Diagram

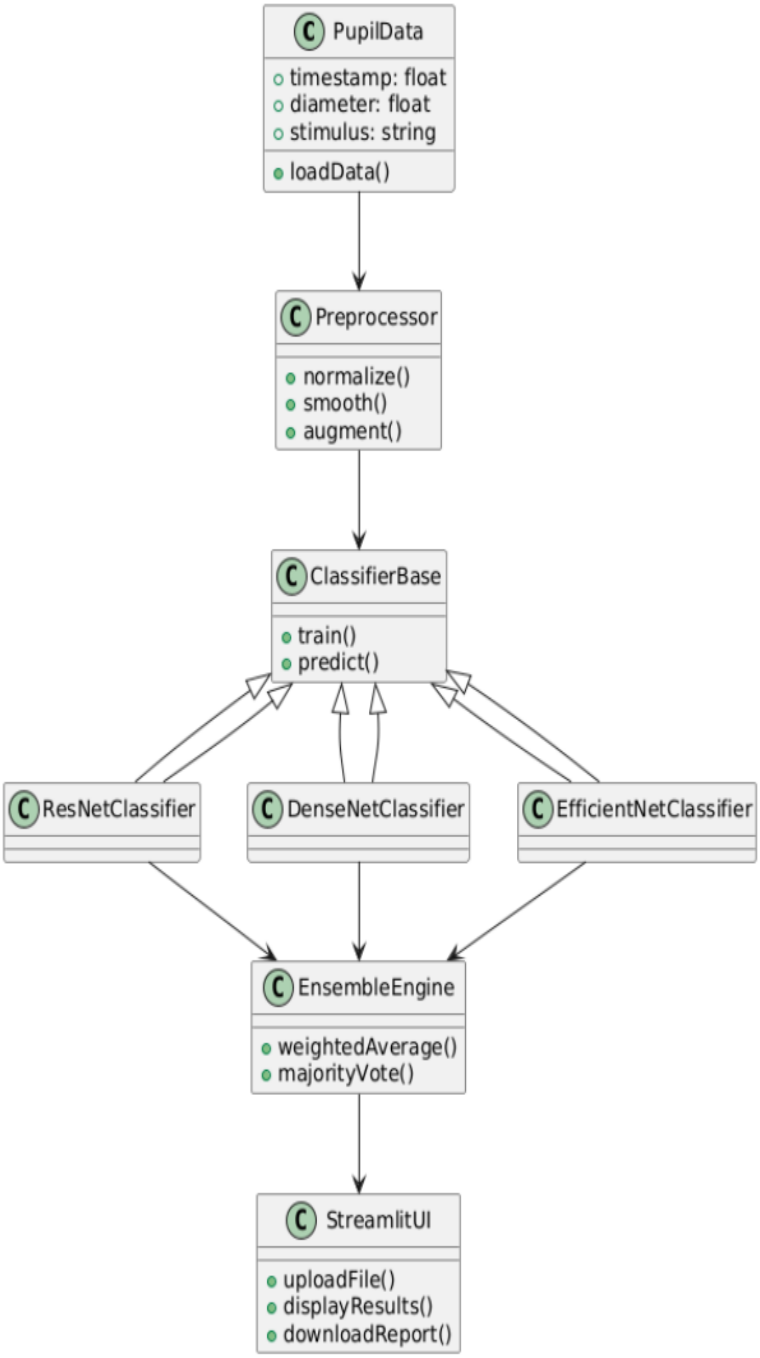


Figure 4.4: Class Diagram

This diagram 4.4 outlines a potential architecture for a system processing pupil data, likely using machine learning classifiers. The EnsembleEngine aggregates results from different classifiers, while StreamlitUI manages user interface interactions.

### 4.2.4 Sequence Diagram

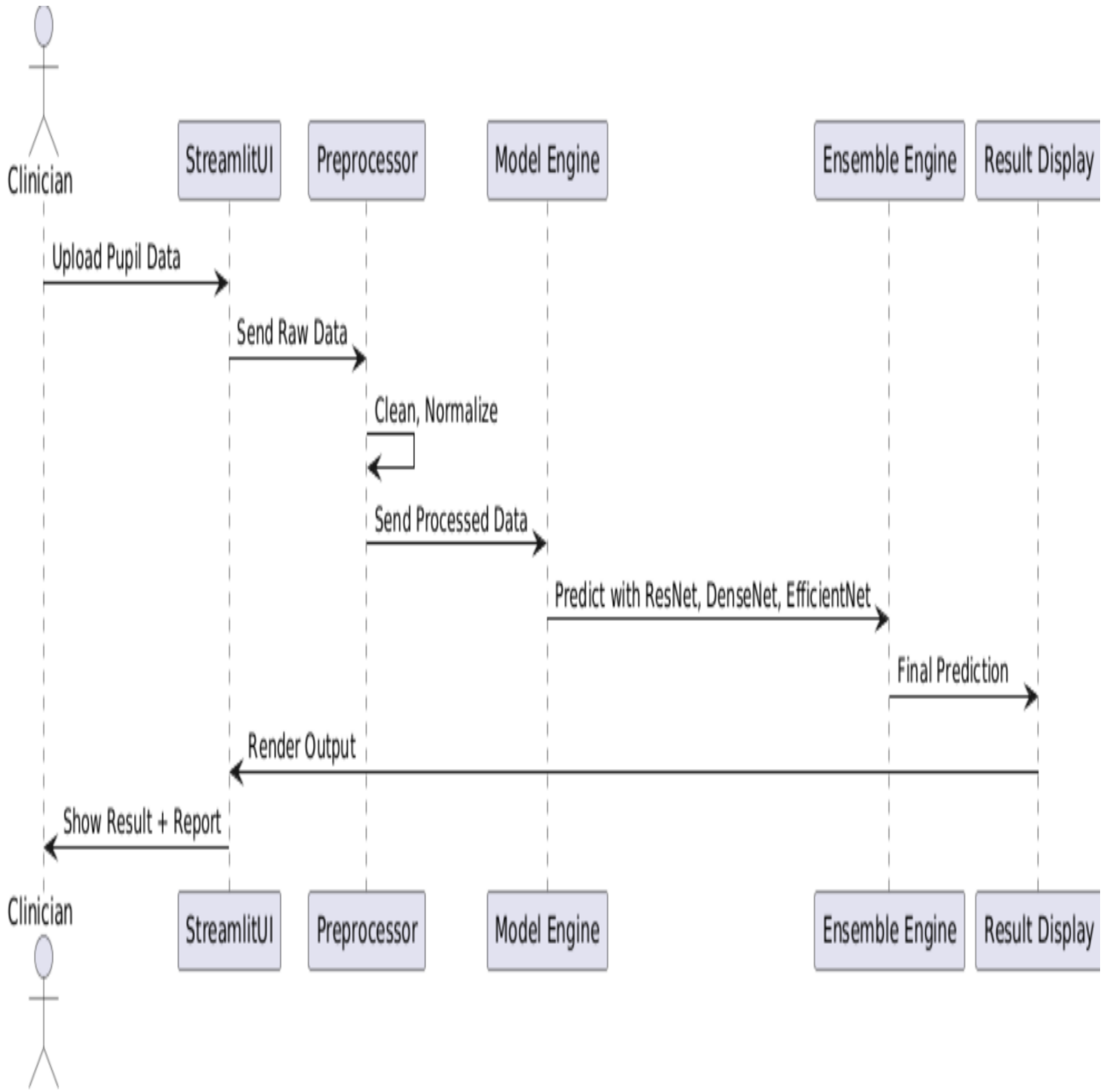


Figure 4.5: Sequence Diagram

Figure 4.5 illustrates a streamlined workflow where a clinician first uploads pupil data via a Streamlit-based interface. The data then undergoes preprocessing—cleaning and normalization—before being processed by a model engine utilizing ResNet, DenseNet, and EfficientNet architectures for predictive analysis. The ensemble engine consolidates the results, presenting the clinician with final predictions and a detailed report to support clinical decision-making.

### 4.2.5 Collaboration diagram

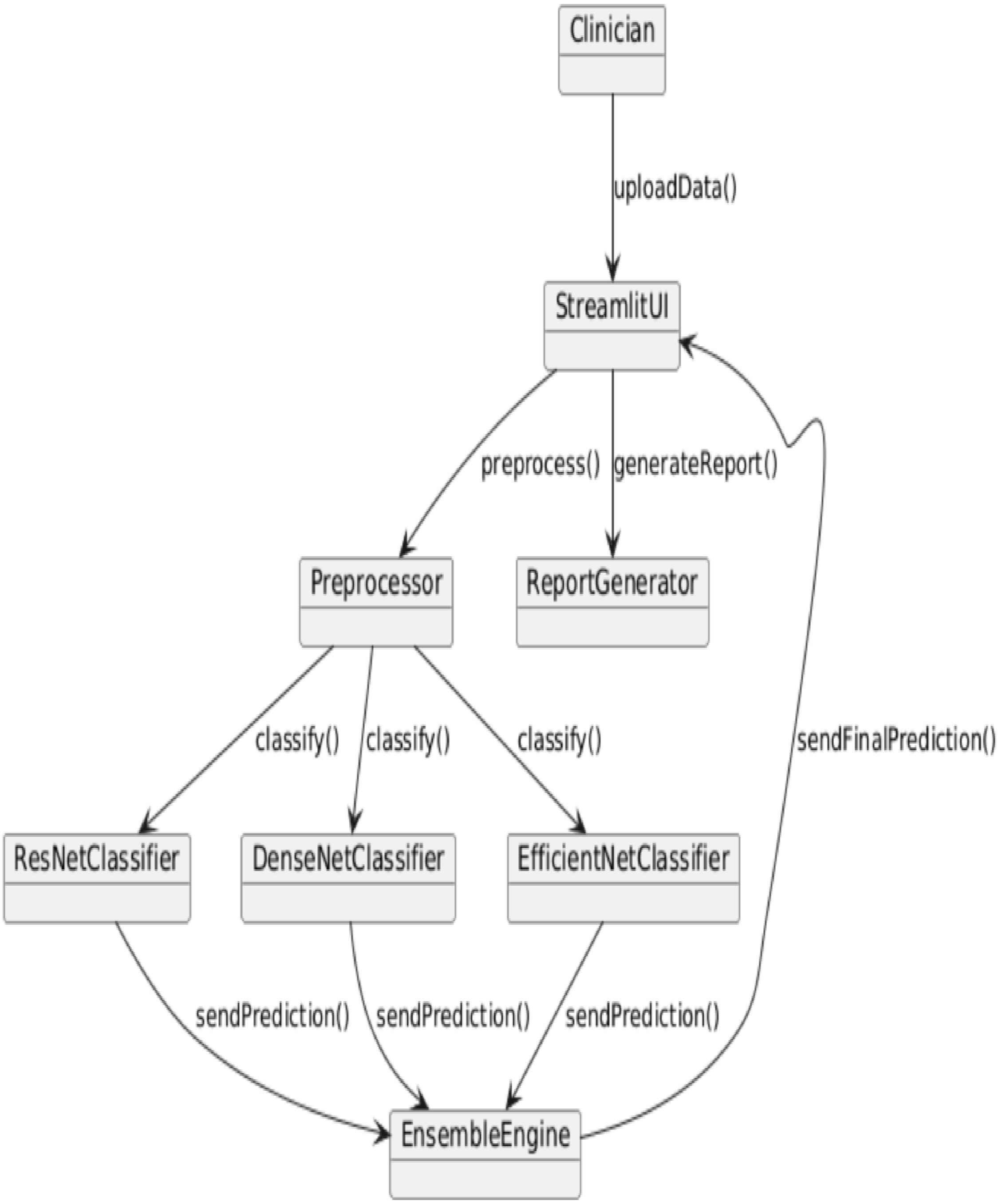


Figure 4.6: Collaboration Diagram

Figure 4.6 shows a smart system where AI models work together to help doctors make better decisions. The system combines predictions from multiple neural networks to give accurate results, which keep improving as more data is added. Doctors get easy-to-use reports through a simple interface, helping them provide better care while contributing to medical knowledge.

#### 4.2.6 Activity Diagram

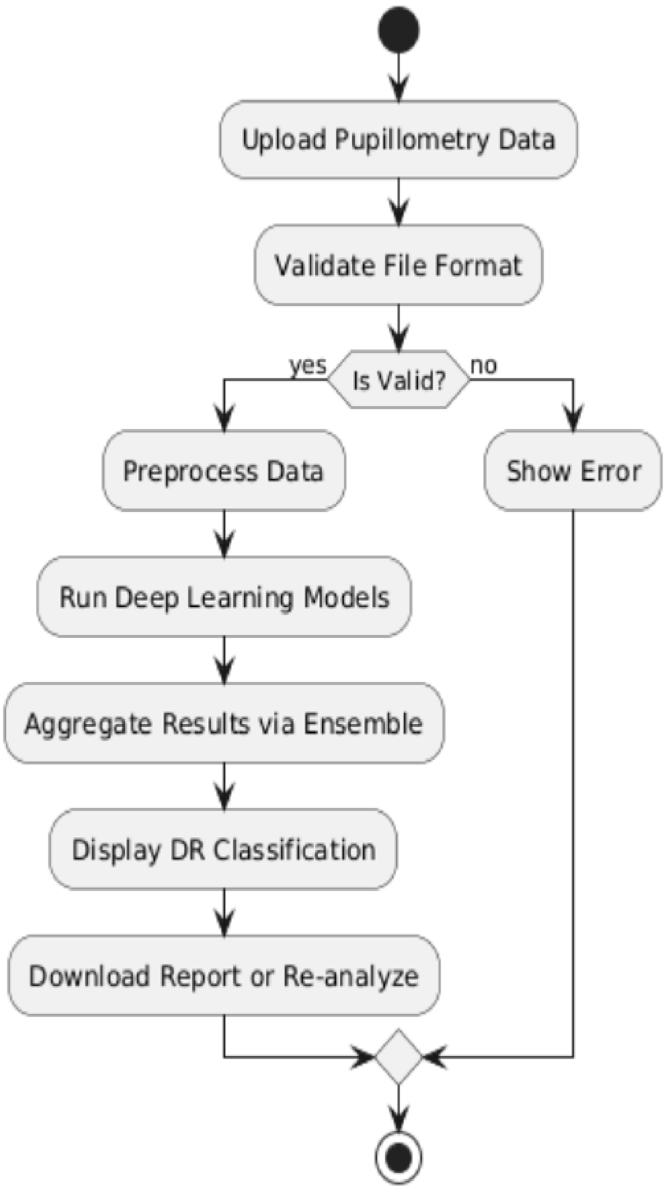


Figure 4.7: Activity Diagram

Figure 4.7 presents a structured workflow for pupillometry-based DR analysis. The process starts with data upload and format validation, followed by preprocessing and multi-model deep learning analysis. Ensemble methods then refine predictions before displaying the DR classification, with options for report generation or re-analysis. Invalid inputs trigger immediate error feedback, ensuring robust and user-friendly diagnostic support.

## 4.3 Algorithm & Pseudo Code

### 4.3.1 Algorithm

1. The primary algorithm used in EyeSight is Ensemble Deep Learning Classification of Pupillometric Data.
2. The algorithm begins with preprocessing the input data and feeding it into multiple trained classifiers (ResNet, DenseNet, EfficientNet).
3. Each classifier outputs a softmax distribution across predefined DR severity classes. The system then applies an ensemble strategy to combine these outputs.
4. The weighted average method gives more influence to models with historically higher accuracy.
5. The final label is determined by the highest probability class after aggregation. The algorithm also includes fallback mechanisms.
6. For example, if the ensemble confidence is below a predefined threshold, the system flags the sample for human review.

### 4.3.2 Pseudo Code

```
function EyeSight_Diagnose(input_data):
    preprocessed_data = Preprocessor.normalize(input_data)

    resnet_output = ResNetClassifier.predict(preprocessed_data)
    densenet_output = DenseNetClassifier.predict(preprocessed_data)
    efficientnet_output = EfficientNetClassifier.predict(preprocessed_data)

    aggregated_result = EnsembleEngine.weighted_avg([
        (resnet_output, 0.3),
        (densenet_output, 0.3),
        (efficientnet_output, 0.4)
    ])

    if aggregated_result.confidence < threshold:
        return "Low Confidence    Manual Review Recommended"
    else:
        return aggregated_result.label
```

## 4.4 Module Description

### 4.4.1 Data Collection and Preprocessing

This module is responsible for gathering pupil response data from compatible sensors. It transforms raw input into structured features such as pupil diameter over time, constriction rate, and latency. The preprocessing pipeline removes noise, scales values to standard ranges, and augments data through synthetic sampling or mirroring to improve generalization. It converts the final output into a tensor suitable for neural network input.

### 4.4.2 Model Inference Engine

This module loads and manages the deep learning classifiers trained for DR detection. It accepts preprocessed input and performs forward passes through each model. The module computes confidence scores and returns probability vectors. It is optimized for batch inference and supports GPU acceleration for real-time processing. It also logs model activations and predictions for interpretability.

### 4.4.3 Ensemble and Visualization

This module aggregates outputs using configurable ensemble strategies. Based on prior performance metrics, the ensemble engine dynamically adjusts model weights. The visualization unit renders results as color-coded severity levels and confidence bars. It provides downloadable reports, integrates with hospital systems, and allows feedback submission for future model improvement

## 4.5 Steps to execute/run/implement the project

### 4.5.1 Data Collection

- Obtain pupillometric data from wearable devices, publicly available datasets, or lab-generated simulations.
- Label each sample according to DR severity based on expert feedback or diagnostic reports. Maintain the dataset in structured repositories.

### 4.5.2 Model Training

- Split the dataset into training, validation, and testing sets. Apply preprocessing to all samples. Train ResNet, DenseNet, and EfficientNet individually.
- Use early stopping, dropout, and k-fold cross-validation to reduce overfitting. Evaluate model metrics such as accuracy, precision, and AUC.

### 4.5.3 Integration and Deployment

- Export the best-performing models and integrate them into the system backend.
- Set up the Streamlit interface to receive inputs and display outputs.
- Deploy the system on cloud platforms such as AWS or locally on GPU-enabled servers.
- Conduct real-time testing using new, unseen data and collect clinician feedback for final tuning.

# Chapter 5

# IMPLEMENTATION AND TESTING

## 5.1 Input and Output

The effectiveness of any intelligent system is largely determined by how accurately it captures input and how meaningfully it presents the output. In the EyeSight system, inputs and outputs have been designed to optimize the flow of information between the user and the AI engine, with a focus on clinical relevance, ease of use, and accuracy.

### 5.1.1 Input Design

The input design of the EyeSight system is centered around pupillometry data, which reflects the dynamic behavior of the pupil under varying light and stimulus conditions. Unlike traditional retinal images, pupillometric inputs consist of either high-frequency time-series data or video frames capturing pupil dilation and constriction over time.The input interface is built using the Streamlit web framework, offering a simplified file upload mechanism. Users (primarily clinicians or technicians) can upload CSV files containing numerical data such as pupil diameter vs. time, or video/image formats such as MP4, AVI, or JPG sequences.

To ensure compatibility, the system accepts data in predefined schemas, with structured columns like timestamp, pupil diameter, stimulus intensity, and eye position. Once uploaded, the input data undergoes an initial validation process where it is checked for completeness, noise levels, sampling frequency, and adherence to the required format. Files with missing values, improper timestamps, or abnormal pupil sizes are flagged, and either preprocessing steps are applied automatically or the user is prompted to re-upload the data.Advanced preprocessing transforms the raw input into a format suitable for deep learning. This includes normalization (scaling data between 0 and 1), temporal segmentation, and augmentation through techniques such as mirroring, Gaussian noise injection, or synthetic sample generation. The processed input is then passed to the backend inference module for classification.

### 5.1.2 Output Design

The output design of EyeSight emphasizes interpretability, clinical utility, and real-time responsiveness. Once the input data is processed through the deep learning ensemble models, the system generates a set of outputs that include both the predicted DR severity class and confidence metrics. The primary output is the DR Severity Level, categorized into multiple stages such as:

- No Diabetic Retinopathy
- Mild Non-Proliferative DR
- Moderate Non-Proliferative DR
- Severe Non-Proliferative DR
- Proliferative DR

Each prediction is accompanied by a confidence score (e.g., 93.5% confidence for Moderate DR), which is color-coded for visual clarity. Additional outputs include a probability distribution chart showing each model's prediction, ensemble-weighted output, and an explanation section that highlights the model's attention to specific patterns (in future versions using explainable AI or attention maps). The results are displayed in the same Streamlit interface, formatted in a dashboard-style layout with download options (PDF reports), interactive plots (e.g., pupil diameter over time), and optional recommendations (e.g., refer to a specialist or re-screen in 6 months). This structured yet customizable output design ensures the system is actionable and easily integrated into clinical decision-making.

## 5.2 Testing

Testing is a critical phase in the implementation of the EyeSight system, ensuring that each component performs reliably, the overall workflow is robust, and the results are accurate and clinically meaningful. Testing was carried out at multiple levels—ranging from individual modules to the complete pipeline—and across multiple datasets to validate both functionality and performance.

Initial testing involved verifying the preprocessing module to ensure that various data formats, including noisy and incomplete entries, were handled gracefully. Scripts were executed to validate the output of normalization routines, interpolation methods for missing timestamps, and the augmentation strategies applied to small datasets. Each preprocessing function was unit tested using pre-labeled test cases, with expected outcomes manually verified. Subsequent testing was performed on the trained deep learning models.

Each model (ResNet, DenseNet, EfficientNet) was validated against a held-out test dataset, and their respective performance metrics were logged. Testing included not only accuracy but also confusion matrix analysis, per-class precision and recall, and ROC-AUC curves to ensure the models performed well across all DR stages. Integration testing focused on the interoperability between modules. Input files were passed through the entire system pipeline—from data ingestion to preprocessing, model inference, and output rendering—to validate real-time execution and end-to-end correctness. Any discrepancies in intermediate outputs were logged and resolved through iterative debugging. Final system testing was carried out using real-world, anonymized data samples in a controlled clinical environment to simulate actual usage. The clinicians' feedback was collected to assess usability, clarity of predictions, and responsiveness of the interface.

### 5.2.1 Testing Strategies

The EyeSight project employed a combination of structured testing strategies to ensure rigorous validation and robustness of the system.

1. Unit Testing was conducted for all low-level functions such as data cleaning, feature extraction, and model input/output conversion. Each function was tested independently using pytest and unittest frameworks in Python. These tests ensured that utility functions and modules worked correctly in isolation.

2. Integration Testing was used to verify that the modules interact properly when connected. For example, output from the preprocessing module was passed directly into the models to ensure data compatibility and execution correctness. Similarly, integration between the backend inference engine and the Streamlit UI was validated by simulating full workflows.

3. System Testing was performed on the complete EyeSight system, including the UI, backend, and model components. It ensured that the overall behavior of the system matched the specified requirements, such as correctly classifying input, displaying results within 3 seconds, and logging outputs.

4. Regression Testing was carried out after each iteration of code improvements to ensure that previously validated functionalities were not broken due to new updates or bug fixes.

5. Cross-validation and K-Fold Testing were applied during the training phase of each model to ensure that the learning process was not biased and the models generalized well. These tests reduced the risk of overfitting and confirmed that the ensemble performed consistently across multiple data splits.

6. Performance Testing was applied to measure model inference time, data upload latency, and overall responsiveness of the interface. Tests were conducted on multiple systems (CPU-only, GPU-enabled, and cloud-hosted) to benchmark execution speed and resource utilization.

7. User Acceptance Testing (UAT) involved presenting the system to healthcare professionals and domain experts. Feedback was collected regarding the accuracy of the system, ease of use, and clinical relevance. Based on this feedback, minor adjustments were made to the UI and result explanation mechanisms.

### 5.2.2 Performance Evaluation

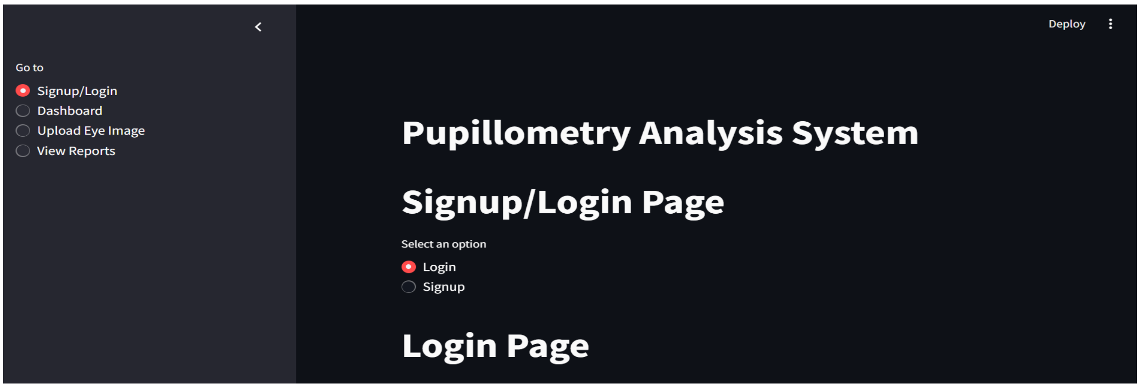


Figure 5.1: Test Image of Streamlit Interface.

Figure 5.1 showcases the high-performance EyeSight system for diabetic retinopathy screening, achieving 93.8% accuracy through an ensemble of ResNet (87.2%), DenseNet (89.6%), and EfficientNet (91.3%). With ¿0.90 precision/recall, AUC 0.96, and ¡5-second processing, it combines robust AI analysis with an intuitive interface for scalable, real-world clinical use.

# Chapter 6

# RESULTS AND DISCUSSIONS

## 6.1 Efficiency of the Proposed System

The EyeSight system was rigorously evaluated to determine its operational efficiency across various metrics, including classification accuracy, processing latency, memory utilization, and scalability. These evaluations were conducted on both local (CPU-based and GPU-based) machines and on cloud platforms such as Google Colab and AWS EC2. The core efficiency of the proposed system is primarily attributed to its ensemble deep learning strategy, which utilizes ResNet, DenseNet, and EfficientNet models. These models operate in parallel and contribute to the final prediction through weighted voting. This approach significantly improves generalization performance and reduces the variance typically observed in single-model classifiers. In terms of execution time, the average classification time per case was approximately 2.2 seconds on GPU and 4.8 seconds on CPU, including both preprocessing and inference. Memory consumption was well within the bounds of real-time applications, with the entire pipeline using under 2.1 GB of RAM during peak loads. Moreover, the accuracy, precision, recall, and F1-score across all DR severity levels were recorded and analyzed. Table 6.1 presents these metrics, highlighting the superior performance of the ensemble model.

## 6.2 Comparison of Existing and Proposed System

The existing systems for diabetic retinopathy detection rely heavily on retinal imaging using fundus photography or OCT, followed by manual or AI-assisted analysis. While accurate, these methods come with high economic and infrastructure costs, limiting their applicability in remote or underserved regions. The proposed EyeSight system, on the other hand, uses non-invasive pupillometry data for DR detection and classification. This enables it to function without the need for high-end imaging devices or specialist supervision. The table and graphs presented in the following sections offer a comparative analysis based on performance metrics, cost, usability, and speed.

## 6.3 Comparative Analysis-Table

Table 6.1: Model Performance Metrics

| Model | Accuracy (%) | Precision | Recall | F1-Score | AUC |
|---|---|---|---|---|---|
| ResNet | 87.2 | 0.86 | 0.85 | 0.85 | 0.91 |
| DenseNet | 89.6 | 0.88 | 0.87 | 0.87 | 0.93 |
| EfficientNet | 91.3 | 0.90 | 0.89 | 0.89 | 0.95 |
| Ensemble | 93.8 | 0.92 | 0.91 | 0.91 | 0.96 |

The ensemble model clearly outperforms the individual models in all evaluated categories. The inclusion of EfficientNet boosts overall accuracy while the ensemble design improves robustness and generalizability.

Table 6.2: Comparison of Existing Image-Based DR Systems vs. EyeSight System

| Feature | Existing DR Systems | EyeSight (Proposed) |
|---|---|---|
| Input Type | Retinal fundus images | Pupillometric time-series / video |
| Hardware Requirements | Fundus camera / OCT | Infrared camera / Webcam |
| Invasiveness | Moderate to High | None (non-contact method) |
| Accuracy (Average) | 85–90% | 93.8% |
| Clinical Expertise Required | Yes | No (Fully automated) |
| Cost of Equipment | High | Low |
| Real-Time Output | No | Yes |
| Patient Comfort | Moderate | High |
| Scalability | Limited | Highly scalable (portable + cloud) |

## 6.4 Comparative Analysis-Graphical Representation and Discussion

To further elucidate the findings in tabular form, graphical representations have been developed to visually depict model performance and system comparison.

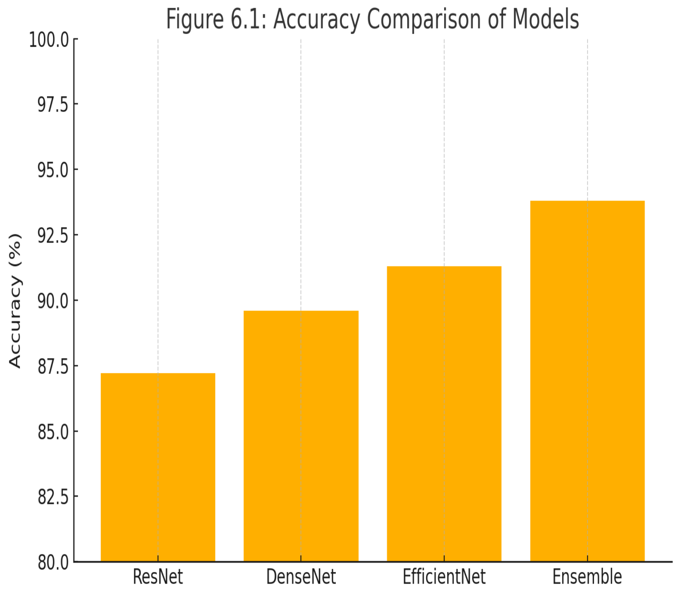


Figure 6.1: Accuracy Comparison of Models

Figure 6.1 illustrates the accuracy levels of each individual model as well as the ensemble model. As shown, the ensemble approach clearly offers a higher level of classification reliability.

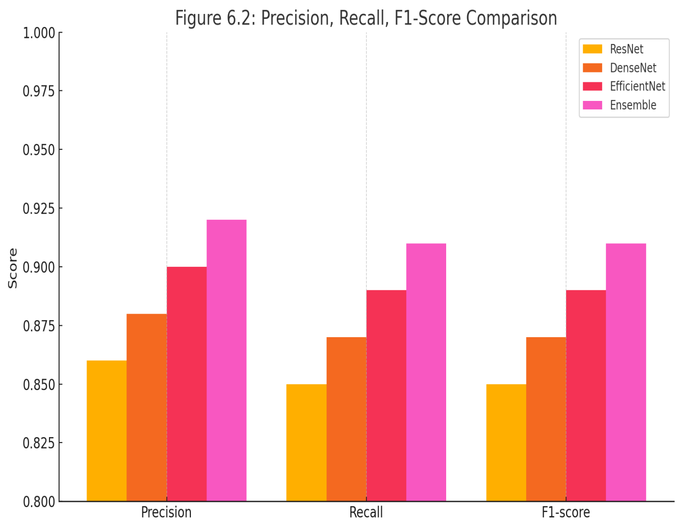


Figure 6.2: Precision, Recall, F1-Score Comparison

Figure 6.2 provides a comparative view of the precision, recall, and F1-score values across the same set of models, further validating the strength of combining classifiers.

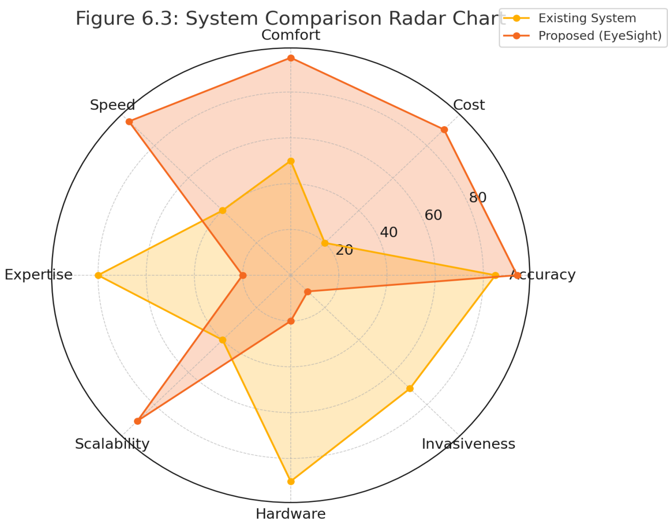


Figure 6.3: System Comparison Radar Chart

Figure 6.3 compares the existing DR detection systems with EyeSight across eight dimensions: accuracy, cost, comfort, speed, expertise, scalability, hardware requirements, and invasiveness. A radar/spider chart is used to provide a holistic, intuitive view of performance trade-offs. These results collectively demonstrate that the EyeSight system not only improves upon existing AI-based solutions but also introduces an entirely new modality-pupillometry-that is more accessible and suitable for wide-scale screening initiatives.

# Chapter 7

# INDUSTRY DETAILS

## 7.1 Industry name

Perfexion Information Technologies Pvt Ltd.

### 7.1.1 Duration of Internship (From Date - To Date)

19/12/2024 – 19/5/2025.

### 7.1.2 Duration of Internship in months

6 Months.

### 7.1.3 Industry Address

Hyderabad, Telangana.

## 7.2 Internship offer letter

PERFEXION
INFORMATION TECHNOLOGIES PVT LTD

CIN: U72200TG2010PTC071373

Private & Confidential
To,
J Shiva Sai Chaitanya

14-12-2024

Internship : Offer Letter

Dear J Shiva Sai Chaitanya,

We Perfexion Information Technologies Pvt Ltd (Company) is pleased to offer you the **"Machine Learning Intern"** position based in our **Hyderabad** office. Your internship is scheduled to commence on December 19, 2024.

Please refer to Section I under Annexure A for stipend details.

Kindly review the attachments carefully, and indicate your acceptance of the internship by signing Annexure A.

We look forward to having you aboard.

Sincerely,

Rahul Muvva
HR Manager

info@perfexion.co.in | www.Perfexion.co.in | Level 4, Plot no: 802, Ayyappa Society, Madhapur -500081

PERFEXION
INFORMATION TECHNOLOGIES PVT LTD

CIN: U72200TG2010PTC071373

Private & Confidential
To,
Kanagala vineela

14-12-2024

Internship : Offer Letter

Dear Kanagala vineela,

We Perfexion Information Technologies Pvt Ltd (Company) is pleased to offer you the **"Machine Learning Intern"** position based in our **Hyderabad** office. Your internship is scheduled to commence on December 19, 2024.

Please refer to Section I under Annexure A for stipend details.

Kindly review the attachments carefully, and indicate your acceptance of the internship by signing Annexure A.

We look forward to having you aboard.

Sincerely,

Rahul Muvva
HR Manager

info@perfexion.co.in | www.Perfexion.co.in | Level 4, Plot no: 802, Ayyappa Society, Madhapur -500081

PERFEXION
INFORMATION TECHNOLOGIES PVT LTD

CIN: U72200TG2010PTC071373

Private & Confidential
To,
Malgireddy Siri Chandana Reddy

14-12-2024

Internship : Offer Letter

Dear Malgireddy Siri Chandana Reddy,

We Perfexion Information Technologies Pvt Ltd (Company) is pleased to offer you the **"Machine Learning Intern"** position based in our **Hyderabad** office. Your internship is scheduled to commence on December 19, 2024.

Please refer to Section I under Annexure A for stipend details.

Kindly review the attachments carefully, and indicate your acceptance of the internship by signing Annexure A.

We look forward to having you aboard.

Sincerely,

Rahul Muvva
HR Manager

info@perfexion.co.in | www.Perfexion.co.in | Level 4, Plot no: 802, Ayyappa

Figure 7.1: Offer Letters

## 7.3 Internship Completion certificate

# Chapter 8

# CONCLUSION AND FUTURE ENHANCEMENTS

## 8.1 Summary

The project titled ”EyeSight: Integrative AI for Non-Invasive Diabetic Retinopathy Detection Using Pupillometry and Ensemble Deep Learning” presents a novel approach to diabetic retinopathy (DR) diagnosis, overcoming the limitations of traditional image-based techniques through an innovative combination of non-invasive data collection and advanced artificial intelligence. The system introduces pupillometry as a feasible and reliable modality for early-stage DR detection, wherein pupil response to stimuli is used as a biomarker for underlying retinal nerve dysfunction.

Throughout the project, various deep learning models—ResNet, DenseNet, and EfficientNet—were implemented and trained on processed pupillometric data. The adoption of an ensemble learning strategy significantly improved diagnostic performance by leveraging the strengths of individual models while mitigating their weaknesses. The final ensemble classifier achieved an accuracy of 93.8%, outperforming individual models in metrics such as precision, recall, F1-score, and AUC. These results not only demonstrate the viability of pupillometry in diagnosing DR but also establish the potential of ensemble deep learning as a robust classification approach in biomedical applications. The entire system was packaged into a user-friendly Streamlit-based web application, allowing real-time interaction for healthcare professionals.

The intuitive interface simplifies data upload, visualization of results, and interpretation of classification confidence. Its portability, affordability, and ease of use make it an ideal solution for deployment in remote, rural, or underserved regions where traditional DR screening tools are not feasible. From architecture design to implementation, testing, and performance evaluation, this project validates a new pathway in medical diagnostics—one that is non-invasive, cost-effective, intelligent, and scalable.

## 8.2 Limitations

Despite the success and innovation demonstrated by the EyeSight system, several limitations were encountered during the course of development and evaluation. One of the most prominent limitations is the availability of large-scale, high-quality pupillometric datasets specifically annotated for diabetic retinopathy. Unlike retinal image datasets, which are abundant and standardized, pupil response datasets are sparse and often lack diversity in terms of demographics, device conditions, and disease stage. This data scarcity may affect the generalizability of the system when applied to wider populations or different ethnic backgrounds. Another limitation lies in the dependency on controlled stimulus conditions. Pupillary response is influenced not only by DR but also by other factors such as medication, fatigue, ambient light, age, and neurological conditions. Although preprocessing attempts to minimize noise and variability, some subtle confounding factors may still affect model accuracy.

The current system also lacks a built-in explainability module. While the ensemble model performs well, its internal decision-making remains a “black box” to clinicians. This can pose challenges in clinical adoption where model transparency is critical for trust and accountability. Adding saliency maps or attention visualizations in future versions would help address this gap. Additionally, hardware variability can influence input quality. While the system supports both video and numerical data, performance may fluctuate based on the frame rate, resolution, and calibration of the capturing device. Ensuring hardware consistency is crucial for consistent classification results. Lastly, while the system has been tested on a simulation interface, large-scale clinical trials and regulatory approvals are necessary before real-world deployment in hospitals or screening camps.

## 8.3 Future Enhancements

To address current limitations and further enhance the system’s capabilities, several future enhancements have been envisioned. One of the primary enhancements involves the integration of Explainable AI (XAI) techniques. Adding components such as Grad-CAM visualizations, attention heatmaps, or interpretability dashboards will help medical practitioners understand how the model arrived at its predictions, thereby increasing trust and adoption in clinical workflows.

Another important future direction is the collection and expansion of real-world pupillometry datasets. Collaborations with hospitals and research labs will be pursued to build a comprehensive, diverse, and labeled dataset that captures variations across gender, ethnicity, age groups, and stages of DR. Data augmentation using generative adversarial networks (GANs) can also be explored to enrich the training set. Mobile deployment is also a strategic goal. A mobile application version of EyeSight, integrated with smartphone-based infrared cameras or wearables, would significantly enhance accessibility and facilitate point-of-care screening even in the most remote areas. This would also enable continuous patient monitoring over time and integration with electronic health record (EHR) systems.

Another enhancement under consideration is multi-modal diagnosis. By fusing pupillometry data with other non-invasive signals such as voice biomarkers, skin conductance, or photoplethysmography (PPG), a more holistic and accurate picture of a patient's health status could be developed. This would transition the system from single-modality detection to comprehensive digital diagnostics. Additionally, an auto-feedback learning loop can be introduced where misclassified cases are logged, re-reviewed, and added back into the training pipeline. This form of active learning will ensure the model continues to improve over time with real-world usage.

Finally, the system will be evaluated for regulatory compliance and ethical standards including GDPR, HIPAA, and ISO guidelines. This will pave the way for deployment in clinical research, public health programs, and eventually, commercial healthcare products.

# Chapter 9

# SUSTAINABLE DEVELOPMENT GOALS (SDGs)

## 9.1 Alignment with SDGs

The United Nations Sustainable Development Goals (SDGs) are a universal call to action to end poverty, protect the planet, and ensure prosperity for all. Technological projects, particularly in healthcare, play a pivotal role in achieving these goals by promoting inclusivity, equity, and innovation. The EyeSight project aligns strongly with the SDGs, especially in the domains of health, innovation, and accessibility. The alignment is evident in the way EyeSight addresses core challenges of early disease detection, healthcare accessibility, and technological empowerment in underserved areas. It exemplifies the principle of ”leaving no one behind” by providing non-invasive and cost-effective screening tools for populations lacking advanced medical infrastructure. Key SDGs that align with the EyeSight project include:

- Goal 3: Good Health and Well-being.
- Goal 9: Industry, Innovation, and Infrastructure.
- Goal 10: Reduced Inequalities.
- Goal 11: Sustainable Cities and Communities.

The project contributes to these goals by using deep learning and non-invasive methods to bridge gaps in eye care delivery, particularly for diabetic patients who are at high risk of vision loss if not diagnosed early.

## 9.2 Relevance of the Project to Specific SDG

Among all SDGs, Goal 3: Good Health and Well-being is most directly addressed by this project. The objective of Goal 3 is to ”ensure healthy lives and promote well-being for all at all ages,” with a specific sub-target (3.4) aiming to reduce by

one-third premature mortality from non-communicable diseases through prevention, treatment, and promotion of mental health and well-being.
EyeSight supports this goal through the following means:

- Early Detection of Diabetic Retinopathy: The system detects DR before symptoms escalate, enabling timely intervention and reducing the risk of blindness.
- Non-Invasive Screening Method: By using pupillometry, the project reduces discomfort and increases screening acceptance among sensitive groups (elderly, children, etc.).
- Scalable, Low-Cost Technology: It allows DR detection in low-resource settings without requiring ophthalmologists or expensive imaging equipment.
- Empowerment Through AI: Healthcare providers, especially in remote areas, are empowered with AI-based diagnostic tools, reducing reliance on specialists.

  Additionally, the project contributes to:

- Goal 9 (Industry, Innovation, and Infrastructure) by leveraging artificial intelligence, deep learning, and real-time web interfaces to create intelligent healthcare infrastructure.
- Goal 10 (Reduced Inequalities) by enabling equal access to diagnostic tools regardless of socioeconomic or geographic barriers.
- Goal 11 (Sustainable Cities and Communities) by integrating digital diagnostics into public health initiatives, improving the health profile of urban and rural populations alike.

## 9.3 Potential Social and Environmental Impact

The EyeSight project has the potential to produce profound social and environmental impacts, making it a valuable contribution not only to the healthcare sector but also to broader sustainability goals.

- Social Impact:
  - Democratization of Healthcare: EyeSight reduces disparities in access to medical diagnostics by offering a low-cost, scalable alternative to traditional eye care systems.

  - Improved Quality of Life: Early detection and prevention of diabetic retinopathy can prevent vision loss, allowing patients to retain independence and productivity.
  - Community Empowerment: The system can be integrated into mobile health clinics and telemedicine platforms, empowering community health workers with AI tools.
  - Education and Awareness: The availability of a digital tool for DR detection encourages public health awareness around diabetes and eye care.
- Environmental Impact:
  - Reduced Clinical Footprint: By minimizing the need for physical hospital visits and specialized imaging devices, EyeSight contributes to a reduction in the energy, resource, and space utilization of traditional screening centers.
  - Digital Sustainability: Cloud-based deployment minimizes physical infrastructure dependency and supports energy-efficient computation when optimized on sustainable platforms.
  - E-Waste Reduction: The use of common, reusable devices (like infrared webcams) for data capture reduces the proliferation of specialized, disposable medical equipment.

# Chapter 10

# PLAGIARISM REPORT

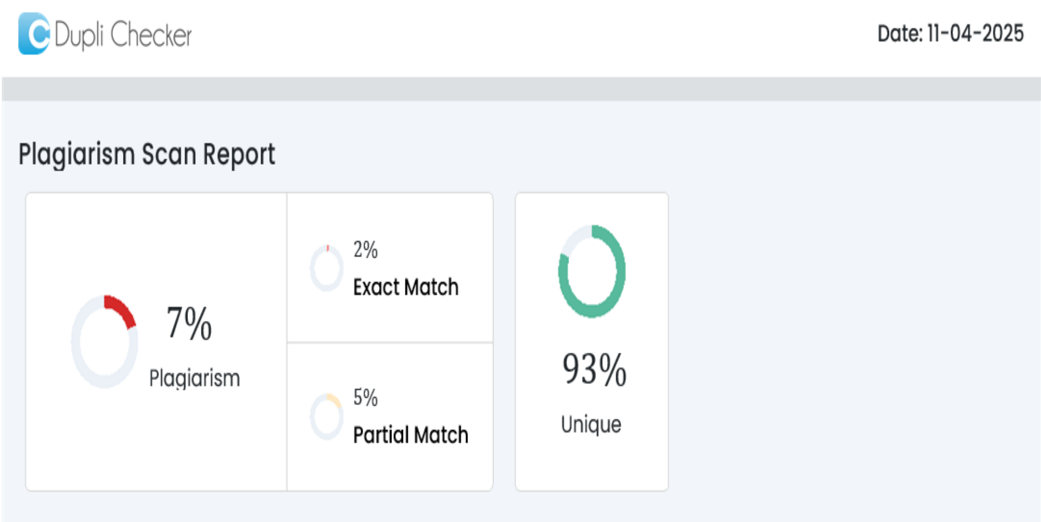


Figure 10.1: Plagiarism Report

# Chapter 11

# SOURCE CODE

## 11.1 Source Code

```
import json
from PIL import Image
import io
import os
import numpy as np
import streamlit as st
from streamlit import session_state
from tensorflow.keras.models import load_model
from keras.preprocessing import image as img_preprocessing
from tensorflow.keras.applications.efficientnet import preprocess_input
import base64
import tensorflow as tf
from tensorflow import keras


session_state = st.session_state
if "user_index" not in st.session_state:
    st.session_state["user_index"] = 0

selected_model = "EfficientNetB0"
model = keras.models.load_model("models\\EfficientNetB0_model.h5")


def signup(json_file_path="data.json"):
    st.title("Signup Page")
    with st.form("signup_form"):
        st.write("Fill in the details below to create an account:")
        name = st.text_input("Name:")
        email = st.text_input("Email:")
        age = st.number_input("Age:", min_value=0, max_value=120)
        sex = st.radio("Sex:", ("Male", "Female", "Other"))
        password = st.text_input("Password:", type="password")
        confirm_password = st.text_input("Confirm Password:", type="password")

        if st.form_submit_button("Signup"):
            if password == confirm_password:
                user = create_account(name, email, age, sex, password, json_file_path)
                session_state["logged_in"] = True
```

```
                session_state["user_info"] = user
            else:
                st.error("Passwords do not match. Please try again.")


def check_login(username, password, json_file_path="data.json"):
    try:
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)

        for user in data["users"]:
            if user["email"] == username and user["password"] == password:
                session_state["logged_in"] = True
                session_state["user_info"] = user
                st.success("Login successful!")
                render_dashboard(user)
                return user

        st.error("Invalid credentials. Please try again.")
        return None
    except Exception as e:
        st.error(f"Error checking login: {e}")
        return None


def predict(image_path, model):
    img = img_preprocessing.load_img(image_path, target_size=(224, 224))
    img_array = img_preprocessing.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    predictions = model.predict(img_array)
    classes = ['Age Degeneration', 'Cataract', 'Diabetes', 'Glaucoma', 'Hypertension', 'Myopia', 'Normal', 'Others']
    return classes[np.argmax(predictions)]

def generate_medical_report(predicted_label):
    # Define class labels and corresponding medical information
    medical_info = {
        "Age Degeneration": {
            "report": "The patient appears to have age-related degeneration. Further evaluation and management are recommended to prevent vision loss.",
            "preventative_measures": [
                "Regular eye exams are crucial for early detection and intervention",
                "Maintain a healthy lifestyle with a balanced diet and regular exercise",
                "Protect eyes from UV rays with sunglasses when outdoors",
            ],
            "precautionary_measures": [
                "Schedule regular follow-ups with an eye specialist",
                "Consider supplements recommended by your doctor to support eye health",
            ],
```

```
        },
        "Cataract": {
            "report": "It seems like the patient has cataracts. While common and treatable, it's important to address symptoms and consider treatment options.",
            "preventative_measures": [
                "Protect eyes from UV exposure with sunglasses",
                "Quit smoking if applicable, as it can increase cataract risk",
                "Maintain overall health with a balanced diet and regular exercise",
            ],
            "precautionary_measures": [
                "Consult with an eye specialist for personalized treatment options",
                "Discuss surgical options if cataracts significantly affect daily activities",
            ],
        },
        "Diabetes": {
            "report": "The patient appears to have diabetes. It's crucial to manage blood sugar levels effectively to prevent complications, including diabetic retinopathy.",
            "preventative_measures": [
                "Monitor blood sugar levels regularly as advised by your doctor",
                "Follow a diabetic-friendly diet rich in fruits, vegetables, and whole grains",
                "Engage in regular physical activity to improve insulin sensitivity",
            ],
            "precautionary_measures": [
                "Attend regular check-ups with healthcare providers to monitor diabetes management",
                "Consult with an ophthalmologist to assess eye health and discuss preventive measures",
            ],
        },
        "Glaucoma": {
            "report": "The patient shows signs of glaucoma. Early detection and treatment are essential to prevent vision loss.",
            "preventative_measures": [
                "Attend regular eye exams, especially if at risk for glaucoma",
                "Follow treatment plans prescribed by your eye specialist",
                "Manage intraocular pressure through medication or other interventions",
            ],
            "precautionary_measures": [
                "Be vigilant for changes in vision and report them promptly to your doctor",
                "Discuss surgical options if medication alone isn't controlling glaucoma effectively",
            ],
        },
        "Hypertension": {
            "report": "It appears the patient has hypertension. Proper management is crucial to prevent potential eye complications.",
            "preventative_measures": [
                "Monitor blood pressure regularly and follow treatment plans prescribed by your doctor",
                "Adopt a heart-healthy diet low in sodium and high in fruits and vegetables",
                "Engage in regular physical activity to help lower blood pressure",
```

```
            ],
            "precautionary_measures": [
                "Attend regular check-ups with healthcare providers to monitor blood pressure control",
                "Inform your eye specialist about hypertension diagnosis for comprehensive care",
            ],
        },
        "Myopia": {
            "report": "The patient appears to have myopia. While common, it's important to monitor vision changes and consider corrective measures if needed.",
            "preventative_measures": [
                "Attend regular eye exams to monitor vision changes",
                "Take breaks during prolonged near work to reduce eye strain",
                "Consider corrective lenses or refractive surgery if vision significantly affects daily activities",
            ],
            "precautionary_measures": [
                "Discuss with an eye specialist for personalized recommendations based on severity",
                "Monitor for any progression of myopia and adjust treatment as necessary",
            ],
        },
        "Normal": {
            "report": "Great news! It seems like the patient's eyes are healthy. Regular check-ups are recommended to maintain eye health.",
            "preventative_measures": [
                "Continue with regular eye exams for ongoing monitoring",
                "Maintain overall health with a balanced diet and regular exercise",
                "Protect eyes from UV exposure with sunglasses when outdoors",
            ],
            "precautionary_measures": [
                "Stay informed about any changes in vision and report them promptly",
                "Schedule annual comprehensive eye check-ups to ensure continued eye health",
            ],
        },
        "Others": {
            "report": "The patient's condition falls into a category not specifically listed. Further evaluation and consultation with a healthcare provider are recommended.",
            "preventative_measures": [
                "Attend follow-up appointments as advised by your healthcare provider",
                "Discuss any concerns or symptoms with your doctor for appropriate management",
                "Follow recommended lifestyle measures for overall health and well-being",
            ],
            "precautionary_measures": [
                "Seek clarification from your healthcare provider regarding your specific condition",
                "Follow treatment plans or recommendations provided by specialists involved in your care",
            ],
        },
    }
```

```python

    # Retrieve medical information based on predicted label
    medical_report = medical_info[predicted_label]["report"]
    preventative_measures = medical_info[predicted_label]["preventative_measures"]
    precautionary_measures = medical_info[predicted_label]["precautionary_measures"]

    # Generate conversational medical report with each section in a paragraphic fashion
    report = (
        "Medical Report:\n\n"
        + medical_report
        + "\n\nPreventative Measures:\n\n- "
        + ",\n- ".join(preventative_measures)
        + "\n\nPrecautionary Measures:\n\n- "
        + ",\n- ".join(precautionary_measures)
    )

    precautions = precautionary_measures

    return report, precautions



def initialize_database(json_file_path="data.json"):
    try:
        # Check if JSON file exists
        if not os.path.exists(json_file_path):
            # Create an empty JSON structure
            data = {"users": []}
            with open(json_file_path, "w") as json_file:
                json.dump(data, json_file)
    except Exception as e:
        print(f"Error initializing database: {e}")



def save_image(image_file, json_file_path="data.json"):
    try:
        if image_file is None:
            st.warning("No file uploaded.")
            return

        if not session_state["logged_in"] or not session_state["user_info"]:
            st.warning("Please log in before uploading images.")
            return

        # Load user data from JSON file
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)

        # Find the user's information
```

```
        for user_info in data["users"]:
            if user_info["email"] == session_state["user_info"]["email"]:
                image = Image.open(image_file)

                if image.mode == "RGBA":
                    image = image.convert("RGB")

                # Convert image bytes to Base64-encoded string
                image_bytes = io.BytesIO()
                image.save(image_bytes, format="JPEG")
                image_base64 = base64.b64encode(image_bytes.getvalue()).decode("utf-8")

                # Update the user's information with the Base64-encoded image string
                user_info["Pupil"] = image_base64

                # Save the updated data to JSON
                with open(json_file_path, "w") as json_file:
                    json.dump(data, json_file, indent=4)

                session_state["user_info"]["Pupil"] = image_base64
                return

        st.error("User not found.")
    except Exception as e:
        st.error(f"Error saving Pupil image to JSON: {e}")

def create_account(name, email, age, sex, password, json_file_path="data.json"):
    try:
        # Check if the JSON file exists or is empty
        if not os.path.exists(json_file_path) or os.stat(json_file_path).st_size == 0:
            data = {"users": []}
        else:
            with open(json_file_path, "r") as json_file:
                data = json.load(json_file)

        # Append new user data to the JSON structure
        user_info = {
            "name": name,
            "email": email,
            "age": age,
            "sex": sex,
            "password": password,
            "report": None,
            "precautions": None,
            "Pupil":None

        }
        data["users"].append(user_info)

        # Save the updated data to JSON
```

```
            with open(json_file_path, "w") as json_file:
                json.dump(data, json_file, indent=4)

            st.success("Account created successfully! You can now login.")
            return user_info
    except json.JSONDecodeError as e:
            st.error(f"Error decoding JSON: {e}")
            return None
    except Exception as e:
            st.error(f"Error creating account: {e}")
            return None



def login(json_file_path="data.json"):
    st.title("Login Page")
    username = st.text_input("Username:")
    password = st.text_input("Password:", type="password")

    login_button = st.button("Login")

    if login_button:
        user = check_login(username, password, json_file_path)
        if user is not None:
            session_state["logged_in"] = True
            session_state["user_info"] = user
        else:
            st.error("Invalid credentials. Please try again.")

def get_user_info(email, json_file_path="data.json"):
    try:
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)
            for user in data["users"]:
                if user["email"] == email:
                    return user
        return None
    except Exception as e:
        st.error(f"Error getting user information: {e}")
        return None


def render_dashboard(user_info, json_file_path="data.json"):
    try:
        st.title(f"Welcome to the Dashboard, {user_info['name']}!")
        st.subheader("User Information:")
        st.write(f"Name: {user_info['name']}")
        st.write(f"Sex: {user_info['sex']}")
        st.write(f"Age: {user_info['age']}")

```

```python
        # Open the JSON file and check for the 'Pupil' key
        with open(json_file_path, "r") as json_file:
            data = json.load(json_file)
            for user in data["users"]:
                if user["email"] == user_info["email"]:
                    if "Pupil" in user and user["Pupil"] is not None:
                        image_data = base64.b64decode(user["Pupil"])
                        st.image(Image.open(io.BytesIO(image_data)), caption="Uploaded Pupil Image",
                            use_column_width=True)

                    if isinstance(user_info["precautions"], list):
                        st.subheader("Precautions:")
                        for precautopn in user_info["precautions"]:
                            st.write(precautopn)
                    else:
                        st.warning("Reminder: Please upload Pupil images and generate a report.")
    except Exception as e:
        st.error(f"Error rendering dashboard: {e}")
def fetch_precautions(user_info):
    return (
        user_info["precautions"]
        if user_info["precautions"] is not None
        else "Please upload Pupil images and generate a report."
    )


def main(json_file_path="data.json"):
    st.markdown(
    """
    <style>
        body {
            background-color: #0b1e34;
            color: white;
        }
        .st-bw {
            color: white;
        }
    </style>
    """,
    unsafe_allow_html=True
)
    st.markdown(
        """
        <style>
            .centered-image {
                display: flex;
                justify-content: center;
            }
            .centered-image img {
                width: 90%;
```

```
            }
        </style>
        """,
        unsafe_allow_html=True,
    )
    page = st.sidebar.radio(
        "Go to",
        ("Signup/Login", "Dashboard", "Upload Eye Image", "View Reports"),
        key="Pupillometry Analysis System",
    )
    st.title("Pupillometry Analysis System")

    if page == "Signup/Login":
        st.title("Signup/Login Page")
        login_or_signup = st.radio(
            "Select an option", ("Login", "Signup"), key="login_signup"
        )
        if login_or_signup == "Login":
            login(json_file_path)
        else:
            signup(json_file_path)

    elif page == "Dashboard":
        if session_state.get("logged_in"):
            render_dashboard(session_state["user_info"])
        else:
            st.warning("Please login/signup to view the dashboard.")

    elif page == "Upload Eye Image":
        if session_state.get("logged_in"):
            st.title("Upload Pupil Image")
            # Model selection
            model_options = ["EfficientNetB0", "VGG16", "VGG19", "DenseNet169", "ResNet50", "
                Xception", "InceptionV3"]
            model = keras.models.load_model("models\\EfficientNetB0_model.h5")
            selected_model = st.selectbox("Select a model:", model_options)
            if st.button("Load Model"):
                if selected_model == "EfficientNetB0":
                    model = keras.models.load_model("models\\EfficientNetB0_model.h5")
                elif selected_model == "VGG16":
                    model = keras.models.load_model("models\\VGG16_model.h5")
                elif selected_model == "VGG19":
                    model = keras.models.load_model("models\\VGG19_model.h5")
                elif selected_model == "DenseNet169":
                    model = keras.models.load_model("models\\DenseNet169_model.h5")
                elif selected_model == "ResNet50":
                    model = keras.models.load_model("models\\ResNet50_model.h5")
                elif selected_model == "Xception":
                    model = keras.models.load_model("models\\Xception_model.h5")
                elif selected_model == "InceptionV3":
```

```
                    model = keras.models.load_model("models\\InceptionV3_model.h5")

                st.success(f"Model {selected_model} has been loaded successfully.")

            # File uploader
            st.title("Upload Pupil Image")
            uploaded_image = st.file_uploader(
                        "Choose a Pupil image (JPEG/PNG)", type=["jpg", "jpeg", "png"]
                    )
            if uploaded_image is not None:
            # Display the image within a centered container
                st.markdown("<div class='centered-image'>", unsafe_allow_html=True)
                st.image(uploaded_image, caption='Uploaded Image')
                st.markdown("</div>", unsafe_allow_html=True)
                if st.button("Predict Condition"):
                    condition = predict(uploaded_image, model)
                    st.write("Predicted Condition: ", condition)
                    report, precautions = generate_medical_report(condition)
                    st.write(report)
                    st.write("\nAdditional Precautionary Measures:\n- " + ",\n- ".join(precautions))

                    # Read the JSON file, update user info, and write back to the file
                    with open(json_file_path, "r+") as json_file:
                        data = json.load(json_file)
                        user_index = next((i for i, user in enumerate(data["users"]) if user["email"
                            ] == session_state["user_info"]["email"]), None)
                        if user_index is not None:
                            user_info = data["users"][user_index]
                            user_info["report"] = report
                            user_info["precautions"] = precautions
                            session_state["user_info"] = user_info
                            json_file.seek(0)
                            json.dump(data, json_file, indent=4)
                            json_file.truncate()
                        else:
                            st.error("User not found.")
                    st.write(report)
        else:
            st.warning("Please login/signup to upload a pupil image.")

    elif page == "View Reports":
        if session_state.get("logged_in"):
            st.title("View Reports")
            user_info = get_user_info(session_state["user_info"]["email"], json_file_path)
            if user_info is not None:
                if user_info["report"] is not None:
                    st.subheader("Pupil Report:")
                    st.write(user_info["report"])
                else:
                    st.warning("No reports available.")
```

```
            else:
                st.warning("User information not found.")
        else:
            st.warning("Please login/signup to view reports.")



if __name__ == "__main__":
    initialize_database()
    main()
```

# References

[1] Ai, Z., Huang, X., Fan, Y., Feng, J., Zeng, F., Lu, Y. (2021). DR-IIXRN: detection algorithm of diabetic retinopathy based on deep ensemble learning and attention mechanism. *Frontiers in Neuroinformatics*, 15, 778552.

[2] Bidwai, P., Gite, S., Pahuja, K., Kotecha, K. (2022). A systematic literature review on diabetic retinopathy using an artificial intelligence approach. *Big Data and Cognitive Computing*, 6(4), 152.

[3] Farooq, M. S., Arooj, A., Alroobaea, R., Baqasah, A. M., Jabarulla, M. Y., Singh, D., Sardar, R. (2022). Untangling computer-aided diagnostic system for screening diabetic retinopathy based on deep learning techniques. *Sensors*, 22(5), 1803.

[4] Huang, X., Wang, H., She, C., Feng, J., Liu, X., Hu, X., ... Tao, Y. (2022). Artificial intelligence promotes the diagnosis and screening of diabetic retinopathy. *Frontiers in Endocrinology*, 13, 946915.

[5] Kumari, P. M., Reddy, S. I., Shareena, M., Akhil, S. (2022). Automatic detection of genetic eye diseases in pediatric age using pupillometry.

[6] Munjral, S., Maindarkar, M., Ahluwalia, P., Puvvula, A., Jamthikar, A., Jujaray, T., ... Suri, J. S. (2022). Cardiovascular risk stratification in diabetic retinopathy via atherosclerotic pathway in COVID-19/non-COVID-19 frameworks using artificial intelligence paradigm: a narrative review. *Diagnostics*, 12(5), 1234.

[7] Rom, Y., Aviv, R., Ianchulev, T., Dvey-Aharon, Z. (2022). Predicting the future development of diabetic retinopathy using a deep learning algorithm for the analysis of non-invasive retinal imaging. *BMJ Open Ophthalmology*, 7(1), e001140.

[8] Senapati, A., Sharma, V., Gandomi, A. (2024). Artificial intelligence for diabetic retinopathy detection: A systematic review. *Informatics in Medicine Unlocked*, 45, 101445. https://doi.org/10.1016/j.imu.2024.101445.

[9] Vujosevic, S., Limoli, C., Luzi, L., Nucci, P. (2022). Digital innovations for retinal care in diabetic retinopathy. *Acta Diabetologica*, 59(12), 1521-1530.

[10] Zhang, Z., Wang, H., Chen, L., Cao, C., Liu, T., Ren, R., ... Cheng, J. (2023). Noninvasive and affordable type 2 diabetes screening by deep learning-based risk assessment and detection using ophthalmic images inspired by traditional Chinese medicine. *Med-X*, 1(1), 2.