

**COMSATS University Islamabad,**

**Abbottabad Campus**

***Automated Retinopathy Diagnosis and result interpretation using Explainable AI (XAI)***

***By***

***Abdul Mohiz Khan Tareen CIIT/FA20-BCS-001/ATD***

***Talal Khan CIIT/ FA20-BCS-029/ATD***

***Muhammad Taha Malik CIIT/ FA20-BCS-015/ATD***

***Supervisor  
 Dr. Sardar Khaliq Uz Zaman***

***Bachelor of Science in Computer Science (2020-2024)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

****

**COMSATS University, Islamabad,**

**Abbottabad Campus**

***Automated Retinopathy Diagnosis and result interpretation using Explainable AI (XAI)***

**A project presented to**

**COMSATS Institute of Information Technology, Islamabad, Abbottabad**

**In partial fulfillment**

**of the requirement for the degree of**

***Bachelor of Science in Computer Science (2020-2024)***

***By***

***Abdul Mohiz Khan Tareen CIIT/FA20-BCS-001/ATD***

***Talal Khan CIIT/ FA20-BCS-029/ATD***

***Muhammad Taha Malik CIIT/ FA20-BCS-015/ATD***

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Talal Khan Muhammad Taha Malik

--------------------------- ---------------------------

Abdul Mohiz Khan Tareen

---------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS) “Automated Retinopathy Diagnosis and result interpretation using Explainable AI (XAI)” was developed by   
**Muhammad Taha Malik (CIIT/FA20-BCS-015/ATD)** , **Abdul Mohiz Khan Tareen (CIIT/FA20-BCS-001/ATD)** and **Talal Khan (CIIT/FA20-BCS-029/ATD)** under the supervision of **Dr. Sardar Khaliq Uz Zaman** and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**EXECUTIVE SUMMARY**

This project's goal is the development of an innovative machine learning-based system aimed at automating the detection of retinopathy from retinal images, while at the same time enhancing interpretability through Explainable AI (XAI) techniques. Retinopathy, a severe ocular condition associated with hypertensive and diabetic patients, requires early detection for timely intervention and prevention of irreversible vision impairment. Our primary objective is to train a robust and accurate machine learning model which can process retinal images, classifying different stages of retinopathy, and aid healthcare professionals in making informed decisions. Key tasks include diverse dataset collection, model and training, comprehensive model evaluation, and user-friendly interface development. Furthermore, we intend to integrate XAI techniques such as trainable attention, CAM, Guided Grad-CAM, Grad-CAM++, and multiple Instance Learning to provide good understanding into the model's decision-making process. This project represents a critical step toward improving the early diagnosis of retinopathy and ensuring transparent and interpretable AI-driven healthcare solutions.

**ACKNOWLEDGEMENT**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor Dr. Sardar Khaliq Uz Zaman. Without his personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Talal Khan Muhammad Taha Malik

--------------------------- ---------------------------

Abdul Mohiz Khan Tareen

---------------------------

**ABBREVIATIONS**

|  |  |
| --- | --- |
| SRS | Software Require Specification |
| PC | Personal Computer |
| DFD | Data Flow Diagram |
| SDD | Software Design Description |

Table of Contents

[1. Introduction 1](#_Toc173002270)

[1.1 Brief 1](#_Toc173002271)

[1.2 Relevance to Course Module 1](#_Toc173002272)

[1.3 Project Background 2](#_Toc173002273)

[1.4 Literature Review 2](#_Toc173002274)

[1.5 Analysis from Literature Review 3](#_Toc173002275)

[1.6 Methodology and Software Lifecycle 4](#_Toc173002276)

[**1.6.1.** Rati**onale behind Selected Methodology** 4](#_Toc173002284)

[2. Problem Definition 6](#_Toc173002285)

[2.1. Problem Statement 6](#_Toc173002286)

[2.2. Deliverables and Development Requirements 7](#_Toc173002287)

[3. Requirement Analysis 8](#_Toc173002288)

[3.1. Use Case Diagram 8](#_Toc173002289)

[3.2. Detailed Use Case 9](#_Toc173002290)

[3.3. Functional Requirements 13](#_Toc173002291)

[3.4. Non-Functional Requirements 14](#_Toc173002292)

[3.5. Usability 14](#_Toc173002293)

[3.6. Performance 15](#_Toc173002294)

[4. Design and Architecture 16](#_Toc173002295)

[4.1. System Architecture 16](#_Toc173002296)

[4.2. Data Representation 17](#_Toc173002297)

[4.3. Process Flow/Representation 20](#_Toc173002298)

[5. Implementation 21](#_Toc173002299)

[5.1. Algorithm 21](#_Toc173002300)

[5.1.1. Data Pre-processing 21](#_Toc173002301)

[5.1.2. Model Architecture 21](#_Toc173002302)

[5.1.3. Flask API Endpoints 22](#_Toc173002303)

[5.1.4. Generative AI Integration 23](#_Toc173002304)

[5.1.5. Explainable AI (XAI) Visualization 24](#_Toc173002305)

[5.2. External APIs 25](#_Toc173002306)

[5.3. User Interface 27](#_Toc173002307)

[5.4 Data Pre-processing 38](#_Toc173002315)

[5.4.1. Introduction 38](#_Toc173002316)

[5.4.2. Image Resizing and Cropping 38](#_Toc173002317)

[5.4.3. Brightness Adjustment 38](#_Toc173002318)

[5.4.4. Application of Gaussian Blur 38](#_Toc173002319)

[5.4.5. Addressing Class Imbalance 39](#_Toc173002320)

[5.4.6. Preprocessing Pipeline Implementation 39](#_Toc173002321)

[5.5 Model Exploration and Comparison 40](#_Toc173002322)

[5.5.1. Introduction 40](#_Toc173002323)

[5.5.2. ResNet 40](#_Toc173002324)

[5.5.3. VGG 40](#_Toc173002325)

[5.5.4. Xception 41](#_Toc173002326)

[5.5.5. EfficientNet 41](#_Toc173002327)

[5.5.6. Fine-Tuning EfficientNet 41](#_Toc173002328)

[5.5.7. Custom Callback Functions 41](#_Toc173002329)

[5.5.8. Learning Rate Optimization 42](#_Toc173002330)

[5.5.9. Regularization Techniques 42](#_Toc173002331)

[5.5.10. Selection of Loss Optimizers 42](#_Toc173002332)

[5.5.11. EfficientNet B3 Model Structure 42](#_Toc173002333)

[5.5.12. Detailed Explanation: 43](#_Toc173002334)

[5.6 Implementation: Explainable AI with Grad-CAM 43](#_Toc173002335)

[5.6.1. Introduction 43](#_Toc173002336)

[5.6.2. Grad-CAM Implementation 44](#_Toc173002337)

[Benefits of Grad-CAM 44](#_Toc173002338)

[Additional Class for Random Images 44](#_Toc173002339)

[Integration with API 45](#_Toc173002340)

[Benefits of Our Approach 45](#_Toc173002341)

[6. Testing and Evaluation 46](#_Toc173002342)

[6.1. Manual Testing 46](#_Toc173002343)

[6.1.1. System Testing 46](#_Toc173002344)

[6.1.2. Unit Testing: 46](#_Toc173002345)

[6.1.3. Functional Testing: 48](#_Toc173002346)

[6.1.4. Integration Testing: 49](#_Toc173002347)

[7. Conclusion and Future Work 50](#_Toc173002348)

[7.1. Conclusion 50](#_Toc173002349)

[7.2. Future Work 50](#_Toc173002350)

[**References** 52](#_Toc173002351)

**List of Tables**

[*Table 1 Show the detailed use case registration. 9*](#_Toc171080275)

[*Table 2 Show the detailed use case Upload Image 9*](#_Toc171080276)

[*Table 3 Show the detailed use case View Results 10*](#_Toc171080277)

[*Table 4 Show the detailed use case Login 10*](#_Toc171080278)

[*Table 5 Show the detailed use case Edit Profile 11*](#_Toc171080279)

[*Table 6 Show the detailed use case View Articles 11*](#_Toc171080280)

[*Table 7 Show the detailed use case Log Out 12*](#_Toc171080281)

[*Table 8 Show the functional requirement for User Authentication 13*](#_Toc171080282)

[*Table 9 Show the functional requirement for Image Upload 13*](#_Toc171080283)

[*Table 10 Show the functional requirement for Retinopathy Detection 13*](#_Toc171080284)

[*Table 11 Show the functional requirement for Result Presentation 14*](#_Toc171080285)

[*Table 12 Details of APIs used in the project 25*](#_Toc171080286)

[*Table 13 user workflow 49*](#_Toc171080287)

**Table of Figures**

[*Figure 1 Use Case Diagram* 8](#_Toc171074370)

[*Figure 2 Data Flow Diagram* 17](#_Toc171074371)

[*Figure 3 Process Flow / Activity Diagram* 20](#_Toc171074372)

[*Figure 4 Login Page* 27](#_Toc171074373)

[*Figure 5 Sign up Page* 28](#_Toc171074374)

[*Figure 6 Home Page* 29](#_Toc171074375)

[*Figure 7 Upload Image* 30](#_Toc171074376)

[*Figure 8 Analyze Image Page* 31](#_Toc171074377)

[*Figure 9 View Result Page* 32](#_Toc171074378)

[*Figure 10 XAI Page* 33](#_Toc171074379)

[*Figure 11 Text Result page* 34](#_Toc171074380)

[*Figure 12 Profile Page* 35](#_Toc171074381)

[*Figure 13 Edit Profile Page* 36](#_Toc171074382)

[*Figure 14 Article Detail Page* 37](#_Toc171074383)

# Introduction

Retinopathy is a condition that causes vision impairment.   Diabetic retinopathy can cause the blood vessels in the retina to swell and rapture or cause new abnormal blood vessel growth on the surface of retina. Hypertensive retinopathy can narrow the blood vessels in the retina due to high blood pressure.

Hence therefore resulting in total blindness. Anyone with high blood pressure or who has diabetes is susceptible to the condition of retinopathy. This disease can potentially infect the body's neurological system, with very fatal results.  However, this illness can be cured with the right treatment and early diagnosis. Previously, ophthalmologists had to spend a lot of time manually diagnosing and classifying the severity of this condition, but with the help of our project, we can automate the process. Our project will incorporate machine learning model for detecting the severity of retinopathy and explainable AI, which will make it easier for us to comprehend how our model arrived at this decision. Our project will produce results quickly and accurately, saving time. The field of medicine will benefit greatly from this research.

## Brief

The project focuses on developing an innovative system for automated retinopathy diagnosis using Explainable AI (XAI) techniques. The primary outcome of our work is a machine learning model capable of accurately detecting and classifying retinopathy from retinal images. We utilized the efficient net architecture along with Python programming language and TensorFlow library for efficient model development. Methodologically, we employed Use Case analysis to identify system functionalities and requirements, ensuring a user-centered approach. The report highlights the importance of early retinopathy detection and the potential impact of our system in improving healthcare outcomes. Additionally, it discusses the integration of XAI techniques to enhance model interpretability, addressing concerns about AI transparency in medical diagnostics. Overall, our project aims to provide a reliable, transparent, and accessible solution for retinopathy diagnosis, leveraging advanced AI technologies.

## Relevance to Course Module

Our project on automated retinopathy diagnosis using Explainable AI (XAI) techniques is closely related to several courses studied during our Bachelor of Computer Science (BCS) program.

***Machine Learning:*** We applied concepts learned in machine learning courses to develop the core algorithm for retinopathy detection. This includes understanding different machine learning models, training techniques, and evaluation methods.

***Artificial Intelligence:*** In our project, we have leveraged principles of artificial intelligence, with a particular focus on the concepts of overfitting, underfitting, and regularization. Drawing from the insights acquired through our studies in AI, we have endeavored to enhance the model's ability to generalize its decision-making processes.

***Database Systems:*** The system involves storing and retrieving data of users. Our understanding of database systems helped in designing an efficient data storage and retrieval mechanism, ensuring data integrity and security.

***Software Engineering:*** Principles of software engineering were applied throughout the development lifecycle of our project. This includes requirements gathering, system design, implementation, testing, and deployment phases, ensuring the development of a robust and maintainable software system.

## Project Background

The project focuses on automated retinopathy diagnosis using Explainable AI (XAI) techniques. Retinopathy is a medical condition affecting the retina, often caused by diseases like diabetes and hypertension. It can lead to severe vision impairment if not diagnosed and treated early.

The idea behind our project is to develop a system that can automatically analyse retinal images to detect the presence and severity of retinopathy. This system aims to assist medical professionals in diagnosing retinopathy more efficiently and accurately.

Explainable AI (XAI) techniques are integrated into the system to provide transparency and insight into the decision-making process of the AI model. This ensures that medical practitioners can understand and trust the results produced by the system.

In essence, our project leverages machine learning and AI technologies to automate the diagnosis of retinopathy, thereby enabling early intervention and prevention of permanent vision impairment.

## Literature Review

The literature review for our project on automated retinopathy diagnosis and interpretation using Explainable AI (XAI) encompasses various aspects related to retinal imaging, AI applications in healthcare, and the intersection of machine learning with medical diagnostics. Here's an overview of the key points covered:

**Retinal Imaging Technologies:** The review delves into existing retinal imaging technologies such as fundus photography and optical coherence tomography (OCT). It discusses their utility in capturing detailed images of the retina, which are crucial for diagnosing retinopathy.

**AI in Healthcare:** The literature review highlights the growing role of artificial intelligence in revolutionizing healthcare, particularly in medical imaging and diagnostics. It explores how AI techniques, including deep learning algorithms, have been increasingly employed to analyse medical images and assist healthcare professionals in disease diagnosis.

**Retinopathy Diagnosis:** Current research and developments in automated retinopathy diagnosis are examined, focusing on advancements in computer-aided diagnosis systems. These systems utilize machine learning models trained on large datasets of retinal images to detect and classify retinopathy, enabling faster and more accurate diagnosis.

**Explainable AI (XAI):** The review discusses the importance of XAI techniques in healthcare AI applications, emphasizing the need for transparency and interpretability in AI-driven decision-making. It explores various XAI methods such as Guided Grad-CAM and Grad-CAM++, which provide insights into how AI models arrive at their predictions, enhancing trust and understanding among medical professionals.

**Ethical and Regulatory Considerations:** The literature review also touches upon the ethical and regulatory considerations surrounding AI adoption in healthcare. It addresses issues related to patient privacy, data security, and the responsible deployment of AI technologies in medical settings.

Overall, the literature review provides a comprehensive overview of the current landscape of retinal imaging, AI applications in healthcare, and the significance of XAI in improving diagnostic transparency and trustworthiness. It lays the groundwork for our project by synthesizing relevant research and identifying gaps and opportunities for further exploration.

## Analysis from Literature Review

In the context of our project on automated retinopathy diagnosis using Explainable AI (XAI), the analysis from the literature review sheds light on several key aspects:

**Current Trends in Retinopathy Diagnosis:** The literature review highlights various approaches to retinopathy diagnosis, including traditional methods employed by ophthalmologists and emerging automated systems leveraging machine learning and AI.

**Advancements in AI for Medical Diagnosis:** It discusses the integration of AI, particularly deep learning algorithms, in medical diagnostics, showcasing their potential in improving accuracy and efficiency.

**Role of Explainable AI (XAI):** The analysis emphasizes the significance of XAI techniques in medical applications, enabling transparency and interpretability in AI-driven decision-making processes. This aligns with our project's objective of incorporating XAI to enhance trust and understanding in retinopathy diagnosis.

**Challenges and Opportunities:** Through the literature review, we identify challenges such as dataset diversity, model interpretability, and ethical considerations in AI-based medical diagnosis. This informs our project's approach to address these challenges through comprehensive dataset collection, model evaluation, and the integration of XAI techniques.

**Comparison with Existing Solutions:** Our project aims to build upon existing research by offering a comprehensive automated retinopathy diagnosis system that encompasses hypertensive retinopathy detection and provides clear explanations of diagnostic results. By integrating XAI techniques, we aim to surpass the limitations of previous systems regarding interpretability and transparency.

## Methodology and Software Lifecycle

The development of the automated retinopathy diagnosis system leverages the Agile methodology combined with the Incremental Model. These methodologies were selected due to their compatibility with the project's dynamic nature, need for iterative progress, and requirement for continuous stakeholder engagement.

1. 6. 1. Rati**onale behind Selected Methodology**

The selection of the Agile methodology and Incremental Model for this project is based on several key factors, tailored to the specific requirements and objectives of developing an automated retinopathy diagnosis system using Explainable AI (XAI):

1. **Flexibility and Adaptability:**

* Agile methodology is chosen for its inherent flexibility, allowing the project team to respond quickly to changing requirements and stakeholder feedback.
* As the field of medical diagnostics evolves rapidly, especially with advancements in AI and machine learning, an agile approach ensures that our system remains adaptable to emerging trends and technologies.

1. **Stakeholder Collaboration:**

* Agile principles emphasize close collaboration with stakeholders throughout the development process.
* In the context of healthcare, involving medical professionals, researchers, and end-users is crucial for understanding their needs, validating system functionalities, and ensuring alignment with industry standards and best practices.

1. **Iterative Development:**

* The incremental nature of Agile development aligns well with the complexity of the project, which involves multiple components such as image processing, AI model training, and result interpretation.
* By breaking down the project into smaller, manageable iterations, we can prioritize and address high-priority features and functionalities iteratively, ensuring a more streamlined and efficient development process.

1. **Risk Management:**

* The Incremental Model, a variation of the Agile approach, mitigates project risks by delivering functionality incrementally.
* Early delivery of essential features allows for early validation and feedback, reducing the risk of misalignment with stakeholder expectations and improving overall project visibility and control.

1. **Continuous Improvement:**

* Both Agile and Incremental methodologies promote a culture of continuous improvement and learning.
* Through regular retrospectives and sprint reviews, the project team can identify areas for enhancement, refine processes, and incorporate lessons learned into subsequent iterations, fostering a cycle of continuous improvement and innovation.

1. **Scalability:**

* Agile's iterative cycles include frequent testing phases, ensuring that each increment meets quality standards before progressing.
* Continuous integration and continuous deployment (CI/CD) practices help in early detection of defects, ensuring a robust and reliable system.

1. **Enhanced Quality Assurance:**

* Both Agile and Incremental methodologies promote a culture of continuous improvement and learning.
* Through regular retrospectives and sprint reviews, the project team can identify areas for enhancement, refine processes, and incorporate lessons learned into subsequent iterations, fostering a cycle of continuous improvement and innovation.

1. **User-Centric Design:**

* Agile encourages regular feedback from end-users, ensuring that the system evolves according to user needs and preferences.
* This user-centric design approach enhances user satisfaction and usability of the system.

# Problem Definition

The problem to be addressed by this project revolves around the manual and time-consuming process of diagnosing retinopathy, a condition that can lead to severe vision impairment and blindness if not detected and treated early. The traditional method of diagnosis requires ophthalmologists to visually inspect retinal images and classify the severity of retinopathy, which can be subjective and prone to human error. Moreover, the increasing prevalence of retinopathy due to conditions like diabetes and hypertension calls for a more efficient and accurate diagnostic approach.

The outcome of this project aims to automate the process of retinopathy diagnosis using machine learning techniques, specifically through the implementation of a novel system that integrates Explainable AI (XAI) methods. By leveraging machine learning models trained on retinal images, the system will be able to detect and classify retinopathy severity with high accuracy and speed. Furthermore, the incorporation of XAI techniques will provide transparency into the decision-making process of the AI model, enhancing trust and understanding among medical professionals.

Overall, the project's objective is to revolutionize retinopathy diagnosis by developing a sophisticated software solution that streamlines the diagnostic process, improves accuracy, and facilitates early intervention to prevent irreversible vision loss.

## Problem Statement

The problem statement for this project revolves around the inefficiency and subjectivity of manual retinopathy diagnosis, particularly in the context of diabetic and hypertensive patients. The traditional method of diagnosing retinopathy involves visual inspection of retinal images by ophthalmologists, which is time-consuming and can lead to inconsistencies in severity classification. Additionally, the increasing prevalence of retinopathy due to conditions like diabetes and hypertension underscores the need for a more accurate and scalable diagnostic approach.

The project aims to address this problem by developing an automated system for retinopathy diagnosis using machine learning techniques and Explainable AI (XAI). Specifically, the system will utilize machine learning models trained on retinal images to detect and classify retinopathy severity. The incorporation of XAI methods will enhance the interpretability of the AI-driven diagnosis, thereby increasing trust and understanding among medical professionals.

In summary, the problem statement highlights the need for a more efficient, accurate, and transparent approach to retinopathy diagnosis, particularly in the context of diabetic and hypertensive patients, which the project aims to fulfill through the development of an automated diagnostic system.

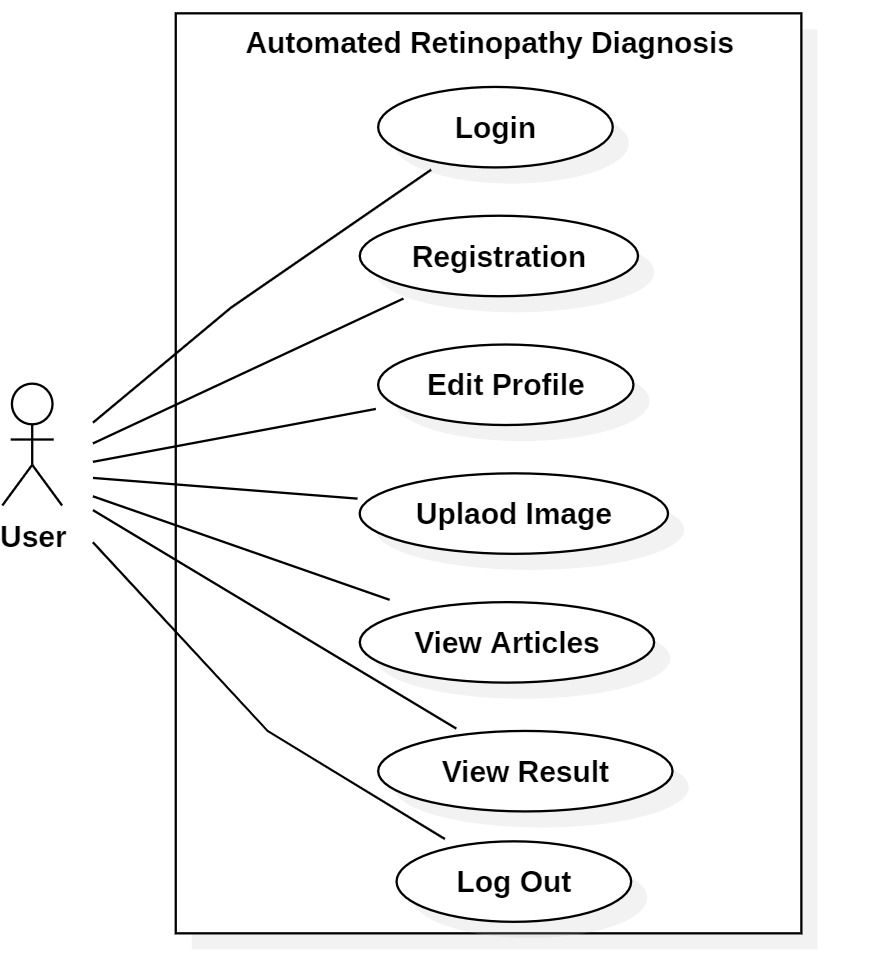
## Deliverables and Development Requirements

The deliverables and development requirements for this project include:

* **Automated Retinopathy Diagnosis System:** The primary deliverable is the development of an automated system capable of accurately detecting and classifying retinopathy severity from retinal images. This system will leverage machine learning algorithms trained on a diverse dataset of retinal images to achieve high diagnostic accuracy.
* **Explainable AI Integration:** Another key deliverable is the integration of Explainable AI (XAI) techniques into the diagnostic system. This will enhance the interpretability of the system's decisions, providing medical professionals with insights into how the diagnosis was reached.
* **User Interface:** A user-friendly interface will be developed to facilitate interaction with the diagnostic system. This interface will allow users to upload retinal images, view analysis results, and interpret diagnostic findings in a clear and understandable format.
* **Scalable Infrastructure:** The project will require scalable infrastructure to support the deployment and operation of the diagnostic system. Cloud services, such as Amazon Web Services (AWS), will be utilized to ensure reliability, scalability, and accessibility of the system.
* **Documentation and Training Materials:** Comprehensive documentation and training materials will be developed to assist users in understanding and utilizing the diagnostic system effectively. This includes user manuals, technical documentation, and training modules for medical professionals.
* **Testing and Evaluation:** Rigorous testing and evaluation procedures will be conducted to validate the accuracy, reliability, and performance of the diagnostic system. This includes testing the system with diverse datasets and evaluating its performance against ground truth diagnoses.

# Requirement Analysis

## Use Case Diagram

****

*Figure 1 Use Case Diagram*

## Detailed Use Case

*Table 1 Show the detailed use case registration.*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1 |
| **Use Case Name** | Registration |
| **Actors** | Primary Actor: User |
| **Description** | A user registers on the platform to create a new account for accessing the retinopathy diagnosis system. |
| **Trigger** | The user initiates the registration process to create a new account. |
| **Preconditions:** | PRE-1. The user is not currently registered on the system.  PRE-2. The platform registration page is accessible. |
| **Postconditions:** | POST-1. The user's account is created, allowing access to platform features upon successful registration. |
| **Normal Flow:** | * User accesses the registration page. * User fills in required details (e.g., username, password, email). * User submits the registration form. * The system validates the information and creates a new user account. * Confirmation message/notification is displayed upon successful registration. |
| **Alternative Flows:]** | - |
| **Exceptions:** | - |
| **Business Rules** | BR-1: Usernames must be unique within the system.  BR-2: Passwords must meet minimum complexity requirements. |

*Table 2 Show the detailed use case Upload Image*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-2 |
| **Use Case Name:** | Upload Image |
| **Actors:** | Primary Actor: User |
| **Description:** | The user uploads fundus images to the platform for analysis of retinal conditions. |
| **Trigger:** | The user initiates the upload process to add fundus images for analysis. |
| **Preconditions:** | PRE-1. The user is logged into their account.  PRE-2. The platform's image upload feature is accessible. |
| **Postconditions:** | POST-1. The uploaded fundus images are ready for analysis by the system. |
| **Normal Flow:** | * User accesses the image upload section. * User selects one or more fundus images from their device. * User initiates the upload process. * System verifies the image file format and size. * Uploaded images are stored in the platform's database. |
| **Alternative Flows:]** | If the selected images do not meet the format or size requirements:  The system prompts the user to select appropriate images. |
| **Exceptions:** | Nonapplicable in this scenario. |
| **Business Rules** | BR-1: Supported image formats include JPEG, PNG.  BR-2: Uploaded images must not exceed a specified file size limit. |

*Table 3 Show the detailed use case View Results*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3 |
| **Use Case Name:** | View Results |
| **Actors:** | Primary Actor: User |
| **Description:** | The user accesses the platform to view the analysis results of the uploaded fundus images. |
| **Trigger:** | The user navigates to the section displaying analysis results. |
| **Preconditions:** | * The user is logged into their account. * Uploaded fundus images have been previously analyzed. |
| **Postconditions:** | POST-1. The user views the analysis results associated with their uploaded images. |
| **Normal Flow:** | * User selects the "View Results" section. * System retrieves and displays the previously analyzed images along with their respective severity assessments. |
| **Alternative Flows:]** | - |
| **Exceptions:** | - |
| **Business Rules** | Results should be displayed in a user-friendly and understandable format. |
|  |  |

*Table 4 Show the detailed use case Login*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4 |
| **Use Case Name:** | Login |
| **Actors:** | Primary Actor: User |
| **Description:** | The user logs into the platform to access their account and use the retinopathy diagnosis system features. |
| **Trigger:** | The user initiates the login process by entering their credentials. |
| **Preconditions:** | PRE-1: The user must have a registered account.  PRE-2: The login page is accessible. |
| **Postconditions:** | POST-1: The user gains access to their account and the platform's features upon successful login. |
| **Normal Flow:** | * User accesses the login page. * Users enter their username and password. * User submits the login form. * The system validates the credentials. * User is redirected to their dashboard upon successful login. |
| **Alternative Flows:]** | If the credentials are incorrect:  The system displays an error message and prompts the user to retry. |
| **Exceptions:** | The system is down or inaccessible. |
| **Business Rules** | BR-1: Passwords must be kept confidential and not shared. |
|  |  |

*Table 5 Show the detailed use case Edit Profile*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5 |
| **Use Case Name:** | Edit Profile |
| **Actors:** | Primary Actor: User |
| **Description:** | The user edits their profile information to update personal details or change account settings. |
| **Trigger:** | The user initiates the edit profile process by accessing their profile settings. |
| **Preconditions:** | PRE-1: The user is logged into their account.  PRE-2: The edit profile page is accessible. |
| **Postconditions:** | POST-1: The user's profile information is updated and saved in the system. |
| **Normal Flow:** | * User accesses the edit profile section. * User updates the desired fields (e.g., email, password, contact information). * User submits the updated profile information. * The system validates and saves the changes. * Confirmation message/notification is displayed upon successful update. |
| **Alternative Flows:]** | If the updated information is invalid:  The system prompts the user to correct the errors. |
| **Exceptions:** | The system is down or inaccessible.  Invalid data entered during the update process. |
| **Business Rules** | BR-1: Usernames cannot be changed once registered.  BR-2: Email addresses must be unique and valid. |
|  |  |

*Table 6 Show the detailed use case View Articles*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6 |
| **Use Case Name:** | View Articles |
| **Actors:** | Primary Actor: User |
| **Description:** | The user views articles related to retinopathy to gain information and insights. |
| **Trigger:** | The user navigates to the section displaying articles. |
| **Preconditions:** | PRE-1: The user is logged into their account.  PRE-2: The articles section is accessible. |
| **Postconditions:** | POST-1: The user views the selected articles. |
| **Normal Flow:** | * User accesses the articles section. * User selects an article to view. * System displays the full content of the selected article. |
| **Alternative Flows:]** | If the article is not available:  The system displays a message indicating the unavailability. |
| **Exceptions:** | The system is down or inaccessible  Selected article fails to load. |
| **Business Rules** | Articles must be accurate and relevant to retinopathy. |
|  |  |

*Table 7 Show the detailed use case Log Out*

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7 |
| **Use Case Name:** | Log Out |
| **Actors:** | Primary Actor: User |
| **Description:** | The user logs out from the platform to end their session. |
| **Trigger:** | The user initiates the log out process by selecting the log out option. |
| **Preconditions:** | PRE-1: The user is logged into their account.  PRE-2: The log out option is accessible. |
| **Postconditions:** | POST-1: The user's session is terminated, and they are redirected to the login page or homepage. |
| **Normal Flow:** | * User selects the log out option. * System terminates the user's session. * User is redirected to the login page or homepage. |
| **Alternative Flows:]** | If the log out process fails:  The system prompts the user to retry. |
| **Exceptions:** | The system is down or inaccessible.  User's session is terminated unexpectedly. |
| **Business Rules** | Users must log out after completing their activities to ensure account security. |
|  |  |

## Functional Requirements

**Functional Requirement 1: User Authentication**

*Table 8 Show the functional requirement for User Authentication*

|  |  |
| --- | --- |
| **Identifier** | REQ-01 |
| **Title** | User Authentication |
| **Requirement** | The system must allow users to register for an account securely and log in securely to access the system. |
| **Source** | User |
| **Rationale** | To ensure secure access to the system and personalized user experience. |
| **Dependencies** | N/A |
| **Priority** | High |

**Functional Requirement 2: Image Upload**

*Table 9 Show the functional requirement for Image Upload*

|  |  |
| --- | --- |
| **Identifier** | REQ-02 |
| **Title** | Image Upload |
| **Requirement** | The system shall provide users with the capability to upload retinal images for analysis. |
| **Source** | User |
| **Rationale** | To allow users to submit retinal images for diagnosis |
| **Dependencies** | N/A |
| **Priority** | High |

**Functional Requirement 3: Retinopathy Detection**

*Table 10 Show the functional requirement for Retinopathy Detection*

|  |  |
| --- | --- |
| **Identifier** | REQ-03 |
| **Title** | Retinopathy Detection |
| **Requirement** | The system should analyse uploaded retinal images to detect the presence and severity of retinopathy. |
| **Source** | System |
| **Rationale** | To assist in the identification and assessment of retinopathy from uploaded images. |
| **Dependencies** | Image Upload functionality (REQ-02) |
| **Priority** | High |

**Functional Requirement 4: Result Presentation**

*Table 11 Show the functional requirement for Result Presentation*

|  |  |
| --- | --- |
| **Identifier** | REQ-04 |
| **Title** | Result Presentation |
| **Requirement** | The system must present the findings of retinopathy detection in a clear and understandable format, including severity rating and diagnosis. |
| **Source** | System |
| **Rationale** | To ensure users can easily interpret and understand the analysis results. |
| **Dependencies** | To ensure users can easily interpret and understand the analysis results. |
| **Priority** | High |

## Non-Functional Requirements

* **Non-Functional Requirement 1: Performance**

**Description:** The system must exhibit efficient performance, processing retinal images and delivering results within a reasonable time frame.

**Metrics:** The system should provide results within 10 seconds for an average-sized retinal image.

* **Non-Functional Requirement 2: Reliability**

**Description:** The system must demonstrate a high level of reliability, ensuring accurate and consistent results in retinopathy severity classification.

**Metrics:** The system should have a maximum error rate of 5% in severity classification across diverse retinal images.

* **Non-Functional Requirement 3: Usability**

**Description:** The user interface must be intuitively designed and user friendly to facilitate easy navigation and interpretation of results for medical professionals

**Metrics:** 90% of users should successfully upload retinal images and interpret results on their initial attempt, reflecting the system's usability.

* **Non-Functional Requirement 4: Security**

**Description:** The system must prioritize the confidentiality and integrity of patient data, implementing robust security measures.

**Metrics:** All user data must be encrypted during transmission and storage. Access to patient records should be restricted to authorized personnel, ensuring a secure environment.

## Usability

* **USE-1:** The system shall allow users to upload retinal images for analysis with a maximum of three interactions.
* **USE-2:** The system interface shall provide clear and concise instructions for image uploading and result retrieval to facilitate ease of learning for first-time users.
* **USE-3:** The system shall incorporate error notifications and offer guidance for users in cases of incorrect or incompatible image file formats during the upload process.
* **USE-4:** The system's image analysis function shall provide results within a maximum of two minutes to maintain user engagement and interaction efficiency.
* **USE-5:** The system's interface shall be designed in compliance with accessibility standards (e.g., WCAG 2.1) to ensure usability for users with disabilities.

## Performance

* **PER-1:** The system's image analysis process should complete within 30 seconds for images up to 5 MB in size, ensuring timely results.
* **PER-2:** Image upload and processing time should not exceed 1 minute per image for images larger than 5 MB but not exceeding 10 MB.
* **PER-3:** The system should maintain a response time of under 2 seconds for user interactions on the interface, such as result retrieval or navigating between pages.
* **PER-4:** The system's server infrastructure should be capable of handling concurrent requests from at least 1000 users without a significant increase in response time or system downtime.
* **PER-5:** The system should maintain an average uptime of 99.9% over a 30-day period, accounting for scheduled maintenance windows and unforeseen outages

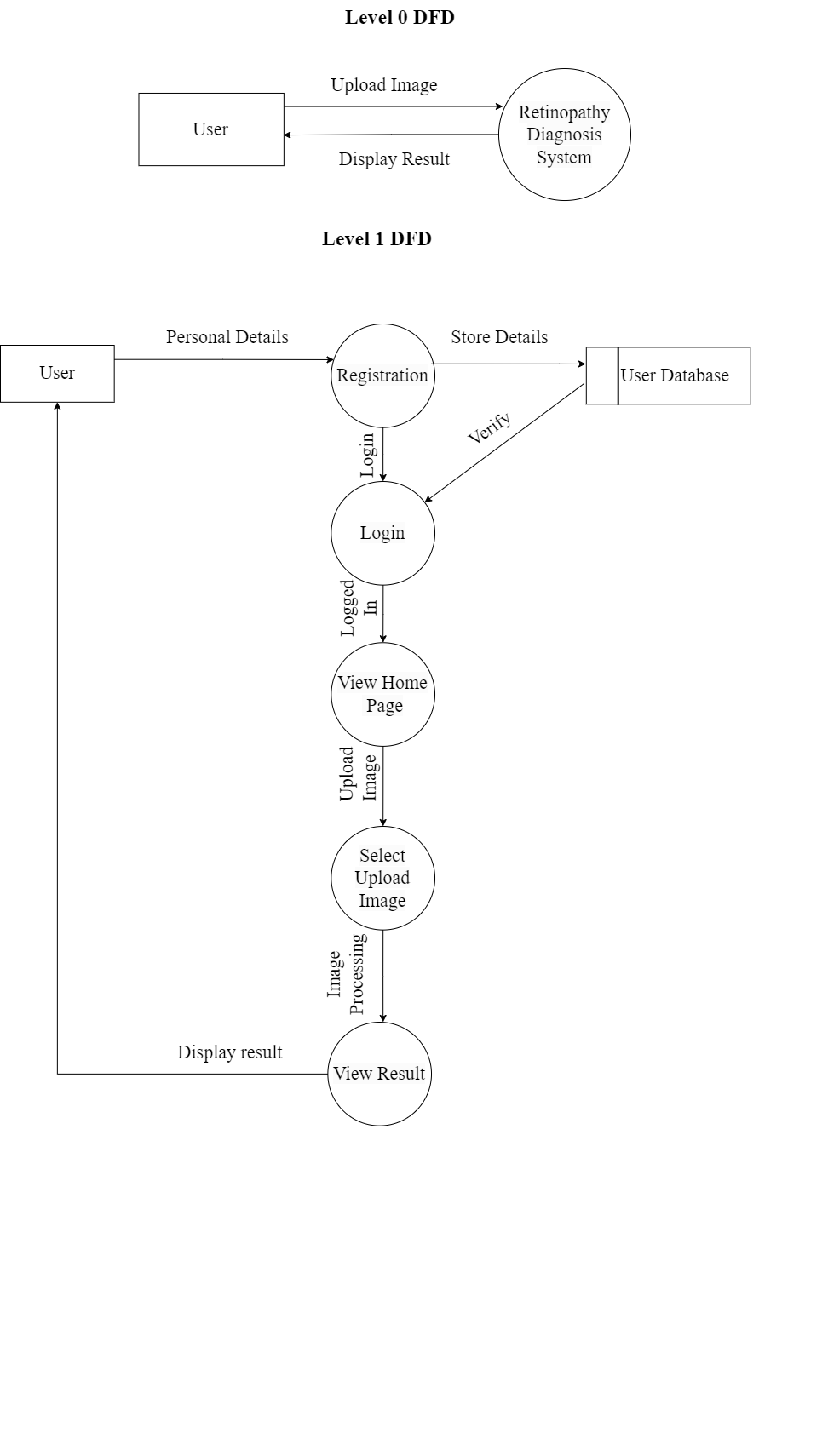
# Design and Architecture

## System Architecture

The system architecture for the project "Automated Retinopathy Diagnosis and Result Interpretation using Explainable AI (XAI)" involves several components:

* User Interface (UI): This component provides the interface for users to interact with the system. It includes features for user registration, image upload, and result visualization.
* Backend Server: The backend server handles user requests, authentication, and serves as the bridge between the user interface and the machine learning models. It manages the processing of uploaded images and communicates with the database.
* Machine Learning Models: This component comprises the core of the system, consisting of machine learning algorithms trained to detect retinopathy from retinal images. The models analyse uploaded images and provide severity assessments.
* Explainable AI (XAI) Module: Integrated with the machine learning models, this module ensures transparency and interpretability in the decision-making process. Techniques like Guided Grad-CAM and Grad-CAM++ are employed to explain the model's predictions.
* Database: The database stores user information, uploaded images, and analysis results. It facilitates data retrieval and management for the system.
* External Services (e.g., Azure): The system utilizes external services like Microsoft App Services for hosting, ensuring scalability, reliability, and security.

## Data Representation



*Figure 2 Data Flow Diagram*

**Data Flow Diagram (DFD) Description:**

The DFD illustrates the flow of data and interactions within the Retinopathy Detection System across two levels.

* **Level 0:**
* ***Entities:*** User and Retinopathy Detection System.
* ***Flow of Data:*** User uploads images to the system, which processes the data and displays the results back to the user.
* **Level 1:**

The process begins with the user interacting with various system components through a sequential flow:

* ***Registration:*** Users register/login to access system functionalities.
* ***Login:*** Verification occurs through the User Database.
* ***View Home Page:*** Users navigate to select options.
* ***Select Upload Image:*** Users initiate image uploads.
* ***View Results:*** Users access results after analysis.
* ***Database Interaction:*** Login process validates against the User Database, while View Results and Compare Results link with user-related data for display and comparison.

**Data design**

* ***User Database:*** Stores user information including usernames, hashed passwords, email addresses, profile images, and other profile details.

**Data dictionary**

**User:**

* **Type:** Entity
* **Description:** Represents individuals registered on the platform.
* **Attributes:**
* user\_id (String): Unique identifier for each user.
* username (String): User's chosen username.
* password (String): Hashed password for user authentication.
* email (String): User's email address.
* profile\_image (String): Path to the profile image uploaded by the user.
* other\_profile\_details (String): Additional profile information provided by the user.

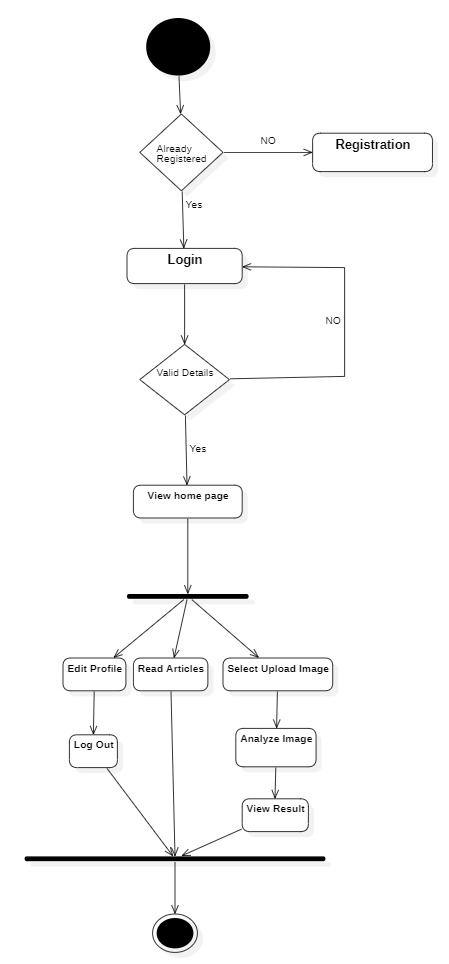
**Fundus Image:**

* **Type:** Entity
* **Description:** Represents the fundus images uploaded by users.
* **Attributes:**
* image\_id (String): Unique identifier for each uploaded image.
* user\_id (String): Identifier linking the image to the user who uploaded it.
* image\_path (String): Location of the uploaded image file.
* upload\_date (DateTime): Date and time when the image was uploaded.

**Analysis Result:**

* **Type:** Entity
* **Description:** Holds the analysis results generated for uploaded fundus images.
* **Attributes:**
* result\_id (String): Unique identifier for each analysis result.
* user\_id (String): Identifier linking the result to the user.
* image\_id (String): Identifier linking the result to the uploaded fundus image.
* analysis\_date (DateTime): Date and time when the analysis was performed.
* severity (String): Severity assessment of the retinopathy condition.
* description (String): Additional descriptive information about the analysis.
* comparison\_data (String): Data used for comparing the results between different analyses.

## Process Flow/Representation



*Figure 3 Process Flow / Activity Diagram*

# Implementation

This chapter details the implementation of the "Diabetic Retinopathy Detection through Retinal Images using Machine Learning" project]/.

## Algorithm

### Data Pre-processing

**Objective:** To prepare raw retinal images for analysis by standardizing size, enhancing quality, and removing irrelevant background information.

**Steps:**

1. Load the raw retinal image.
2. Resize the image to 224x224 pixels.
3. Crop the image to remove unnecessary background using grayscale thresholding.
4. Enhance image contrast and clarity using Gaussian blurring and weighted addition.

**Pseudocode:**

Algorithm PreprocessImage

Input: Raw Retinal Image

Output: Preprocessed Image

resized\_image <- Resize(image, (224, 224))

cropped\_image <- CropImageFromGray(resized\_image, tolerance=7)

enhanced\_image <- EnhanceImage(cropped\_image, sigmaX=10)

return enhanced\_image

**Explanation:** The preprocessing algorithm standardizes the image size for consistent input to the neural network. It then removes unnecessary background to focus on the retinal area. Finally, it enhances the image quality to highlight important features, which is crucial for accurate detection of diabetic retinopathy signs.

### Model Architecture

**Objective:** To create a deep learning model capable of accurately classifying different stages of diabetic retinopathy.

**Steps:**

1. Load pre-trained EfficientNetB3 as the base model.
2. Add custom layers for fine-tuning and classification.
3. Compile the model with appropriate optimizer and loss function.

**Pseudocode:**

Algorithm BuildModel

Input: Image Shape (224, 224, 3), Class Count (6)

Output: Compiled Model

base\_model <- LoadEfficientNetB3(input\_shape=image\_shape, include\_top=False, weights="imagenet")

model <- Sequential([

base\_model,

BatchNormalization(),

Dense(2040, activation='relu', regularization=L1L2),

Dropout(0.45),

Dense(class\_count, activation='softmax')

])

Compile(model, optimizer=Adamax(learning\_rate=0.001), loss='categorical\_crossentropy')

return model

**Explanation:** The model uses EfficientNetB3 pre-trained on ImageNet for efficient feature extraction. Custom layers are added to fine-tune the model for diabetic retinopathy classification. Regularization techniques like Dropout and L1L2 regularization are used to prevent overfitting.

### Flask API Endpoints

**Objective:** To create a user-friendly interface for healthcare professionals to upload retinal images and obtain diagnostic results.

**Steps:**

1. Set up Flask application with necessary routes.
2. Implement image upload and preprocessing functionality.
3. Create prediction endpoint that uses the trained model.
4. Develop explainable AI endpoint for result interpretation.

**Pseudocode:**

Algorithm FlaskApplication

Initialize Flask app

Load trained model

@route '/RetinaAPI/v1/preprocess'

Save uploaded image

preprocessed\_image <- PreprocessImage(uploaded\_image)

Return preprocessed\_image

@route '/RetinaAPI/v1/predict'

preprocessed\_image <- LoadPreprocessedImage()

prediction <- Model.Predict(preprocessed\_image)

Return FormatPrediction(prediction)

@route '/RetinaAPI/v1/xai'

preprocessed\_image <- LoadPreprocessedImage()

xai\_visualization <- GenerateGradCAM(preprocessed\_image, Model)

Return xai\_visualization

Run Flask app

**Explanation:** The Flask application provides a RESTful API for image upload, preprocessing, prediction, and explainable AI visualization. This allows for easy integration with front-end applications or other systems, providing a complete solution for diabetic retinopathy detection.

### Generative AI Integration

**Objective:** To provide detailed, contextual information about different stages of diabetic retinopathy to supplement the model's predictions.

**Steps:**

1. Set up Google's Generative AI model.
2. Create a prompt template for diabetic retinopathy information.
3. Implement an endpoint to generate and return structured content.

**Pseudocode:**

Algorithm GenerateContentEndpoint

Input: Diabetic Retinopathy Stage

Output: Structured Information

Configure GenerativeAI with API key

prompt <- CreatePrompt(stage)

raw\_response <- GenerativeModel.Generate(prompt)

parsed\_content <- ParseJSONResponse(raw\_response)

formatted\_response <- StructureContent(parsed\_content)

Return formatted\_response

**Explanation:** This feature enhances the application by providing AI-generated, detailed information about each stage of diabetic retinopathy. It offers users additional context and educational content, making the tool more comprehensive and informative.

### Explainable AI (XAI) Visualization

**Objective:** To provide visual explanations for the model's predictions, enhancing interpretability and trust in the system.

**Steps:**

1. Generate class activation map using Grad-CAM technique.
2. Resize the activation map to match the original image size.
3. Apply a color map to the activation map for visualization.
4. Overlay the colored activation map on the original image.

**Pseudocode:**

Algorithm GenerateXAIVisualization

Input: Preprocessed Image, Model

Output: XAI Visualization

features, predictions <- Model.PredictWithFeatures(image)

class\_index <- ArgMax(predictions)

grad\_cam <- GenerateGradCAM(features, class\_index)

resized\_grad\_cam <- Resize(grad\_cam, original\_image.shape)

colored\_grad\_cam <- ApplyColorMap(resized\_grad\_cam)

visualization <- OverlayImages(original\_image, colored\_grad\_cam)

Return visualization

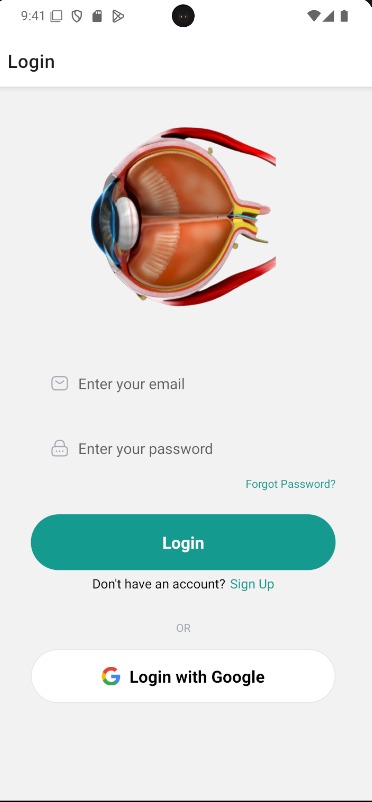
**Explanation:** The XAI visualization helps healthcare professionals understand which parts of the retinal image influenced the model's prediction. This transparency is crucial in medical applications, allowing for verification of the model's focus areas and potentially highlighting areas of concern that require further examination.

## External APIs

*Table 12 Details of APIs used in the project*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name of API** | **Description of API** | **Purpose of usage** | **List down the function/class name in which it is used** |
| Firebase Authentication | Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app | To handle user authentication and store login information securely. | Not explicitly shown in the provided code, but likely used in user management functions |
| Google Gemini API | Gemini APIGoogle's generative AI model that can understand and generate natural language, images, and code. | To generate detailed information about diabetic retinopathy stages based on the classification result. | generate Content function in the Flask app |
|  |  |  |  |
| Keras | Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. | To build and train deep learning models for image analysis and classification of fundus images | build model function  train model function |
| NumPy | NumPy is a fundamental package for scientific computing with Python. | To handle numerical operations, array manipulations, and other mathematical functions | preprocess\_images function. calculate\_metrics function |
| OpenCV | OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. | To process and enhance fundus images before analysis and classification | preprocess\_images function detect\_features function |
| Flask | Flask is a lightweight WSGI web application framework in Python. | To create the web application and handle HTTP requests and responses | app initializationroute functions for various endpoints |
| TensorFlow | TensorFlow is an end-to-end open source platform for machine learning. | To support the backend operations of Keras for building and training deep learning models | backend operations within Keras |
| Pandas | Pandas is a fast, powerful, flexible, and easy-to-use open-source data analysis and data manipulation library for Python. | To handle data manipulation and analysis, especially for organizing and preprocessing the data | data\_cleaning function  data\_analysis function |
| Matplotlib | Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. | To visualize the data and results of the analysis and model predictions | plot\_results function  display\_images function |
| Scikit-learn | Scikit-learn is a machine learning library for Python that provides simple and efficient tools for data mining and data analysis. | To perform machine learning tasks such as data preprocessing, model evaluation, and metrics calculation | evaluate\_model function  split\_data function |

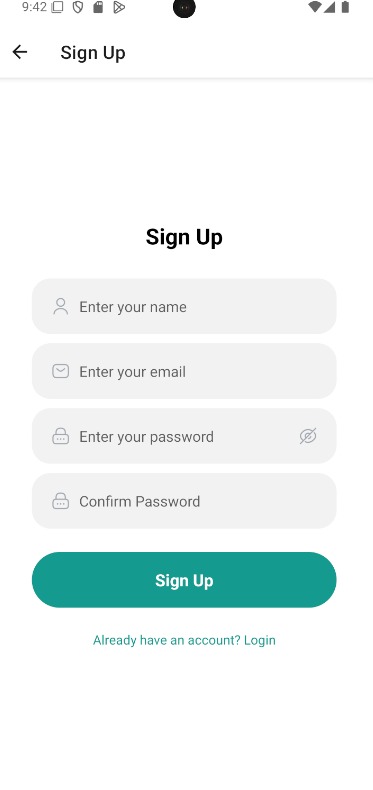
## User Interface



*Figure 4 Login Page*

**Title:** Login Interface

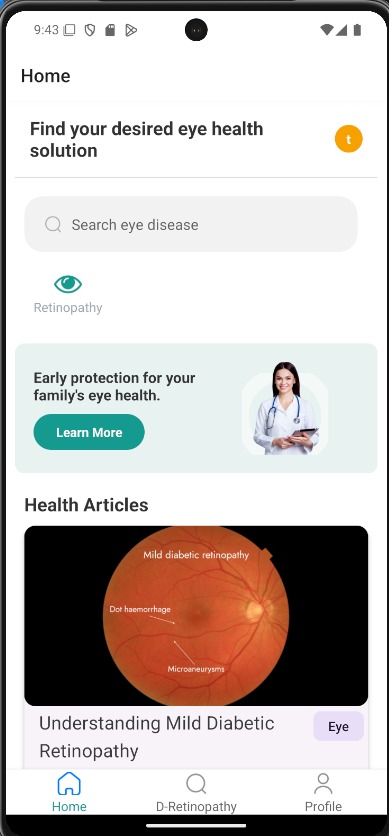
**Description:** The login page welcomes users to sign in to their accounts. Users who already have retinopathy accounts can log in by entering their email and password. There is also a "Forgot Password?" option for password recovery. For new users, there is a "Sign Up" option available. Additionally, users can log in using their Google accounts, which is facilitated by a "Login with Google" button below the main login form.



*Figure 5 Sign up Page*

**Title:** Sign-Up Interface

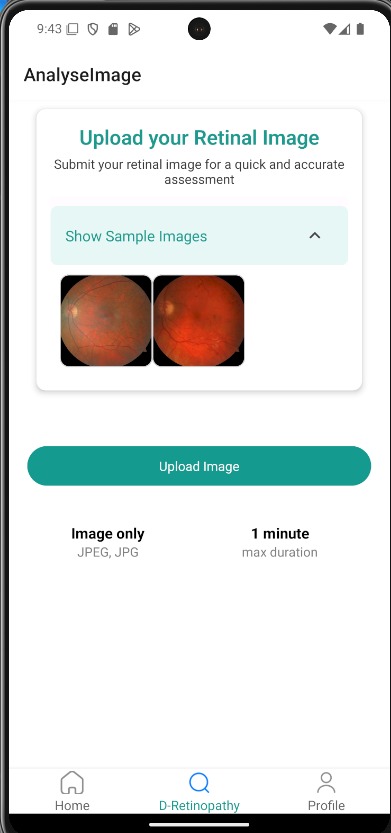
**Description:** The sign-up page allows new users to create an account. Users are prompted to enter their name, email, password, and confirm the password. There is a prominent "Sign Up" button to complete the registration process. For those who already have an account, there is an option to "Login," indicated at the bottom of the page.



*Figure 6 Home Page*

**Title:** Home Page Interface

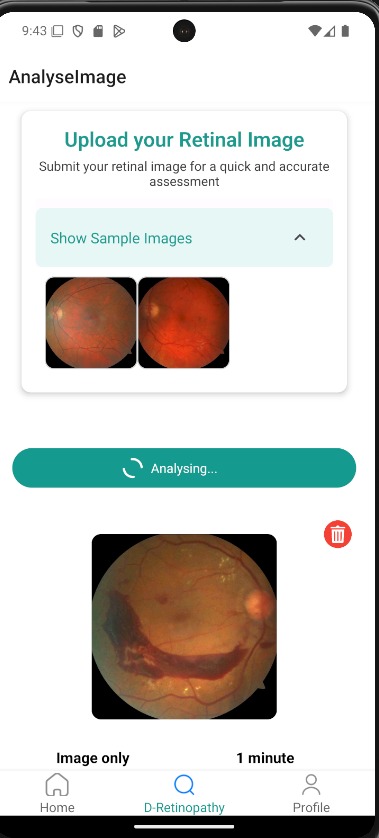
**Description:** The home page provides users with a comprehensive overview of eye health resources. At the top, there is a search bar labeled "Find your desired eye health solution" where users can search for eye diseases and related articles. Below the search bar, there is a section for "Retinopathy" with a call-to-action button labeled "Learn More," highlighting early protection for family eye health. Additionally, the page features a section titled "Health Articles" with a visual example (an image of an eye condition) to educate users about non-diabetic retinopathy.



*Figure 7 Upload Image*

**Title:** Upload Image

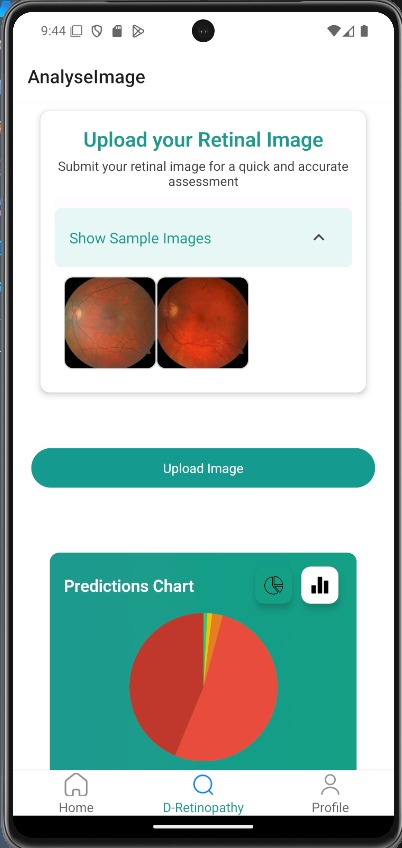
**Description:** This is the upload fundus image page, where users can upload images of their eye fundus for analysis. The interface guides users to select and upload a clear image of their eye, which is crucial for accurate diagnosis and monitoring of retinal health. Users can select an existing image from their gallery. Once the image is uploaded, it will be processed and analyzed for signs of retinopathy, providing users with valuable insights into their eye health.



*Figure 8 Analyze Image Page*

**Title:** Analyze Image

**Description:** Once the image is uploaded, users are directed to the analysis page. Here, the uploaded fundus image is processed and analysed for any signs of retinopathy or other eye conditions. The system utilizes advanced algorithms to detect abnormalities and provide a detailed report on the retinal health of the user. This analysis helps users understand their eye health better and take necessary actions based on the results.

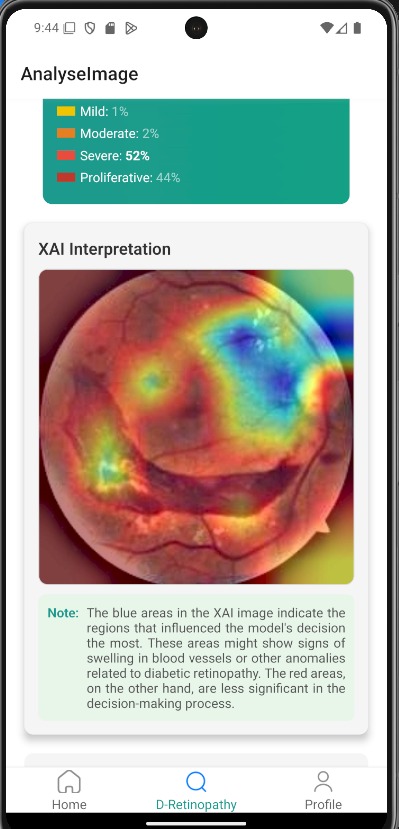


*Figure 9 View Result Page*

**Title:** Upload Retinal Image

**Description:**

This screen allows users to upload their retinal images for analysis. Users can submit their retinal images by clicking the "Upload Image" button, which will then be processed by the system. The page also provides sample images to guide users on what type of images should be uploaded. The uploaded images will undergo a detailed examination to assess the retinal health, detecting any signs of retinopathy or other related conditions.

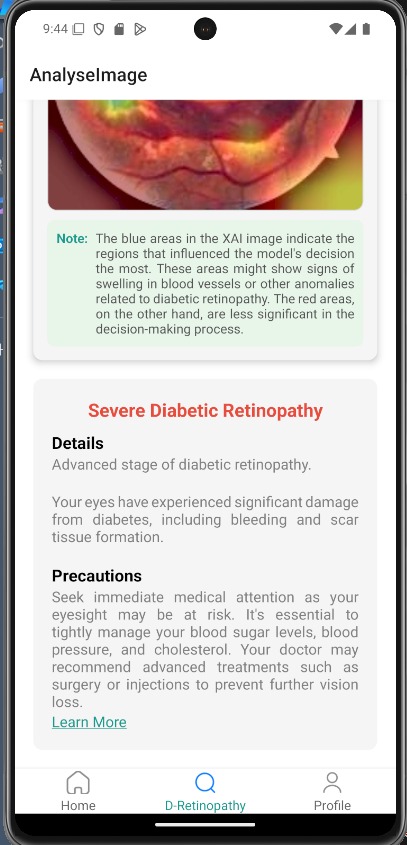


*Figure 10 XAI Page*

**Title:** Predictions Chart and XAI Interpretation

**Description:**

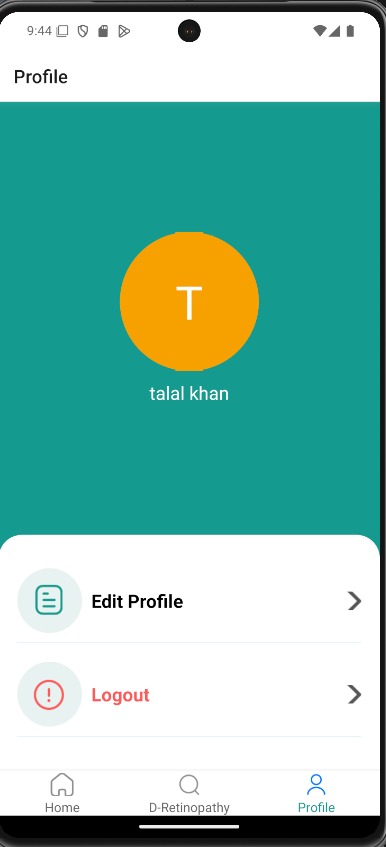
After uploading the retinal image, users are presented with the analysis results. The "Predictions Chart" section displays a pie chart summarizing the severity levels of retinopathy detected, categorized into mild, moderate, severe, and proliferative stages. Below the chart, the "XAI Interpretation" section provides a visual explanation of the AI model's decision-making process. The XAI (Explainable AI) image highlights areas of the retina that influenced the model's diagnosis, with blue regions indicating significant areas of concern and red areas being less influential. This detailed interpretation aids users in understanding the underlying factors contributing to their diagnosis.



*Figure 11 Text Result page*

**Title:** Result

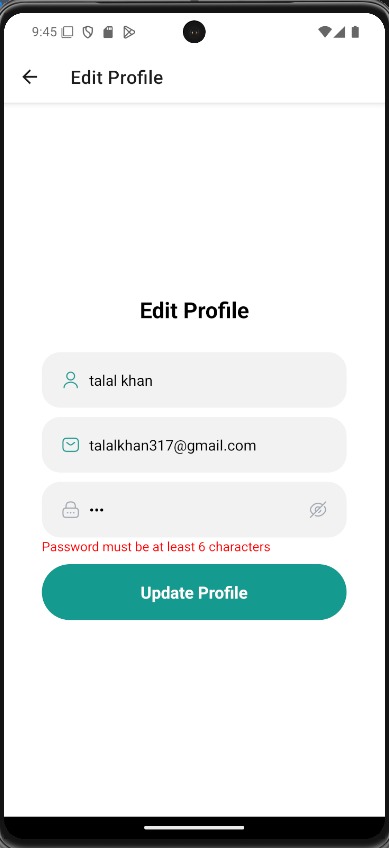
**Description:** After the analysis is complete, the results are displayed to the user in an easy-to-understand format. Key findings from the analysis are presented visually, such as in a pie chart, to clearly show the proportions of different conditions or risk factors detected. This visual representation helps users quickly grasp the overall health status of their eyes and understand the specific areas that may require attention or further consultation with a healthcare Professional



*Figure 12 Profile Page*

**Title:** Profile

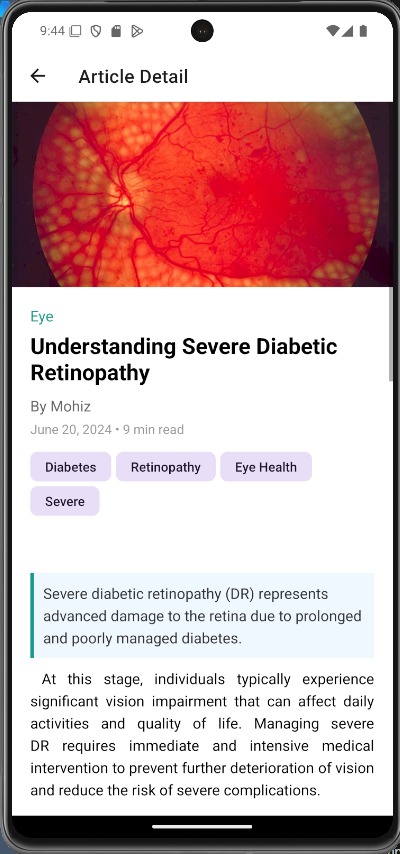
**Description:** The Profile page features a screen title labeled "Profile" and displays the user's profile information, including a profile picture placeholder represented by a large 'T' in an orange circle for the user "Talal Khan." The page offers options such as an "Edit Profile" button, which navigates to the Edit Profile screen, and a "Logout" button for logging out of the account. At the bottom, a footer menu provides navigation buttons for Home, D-Retinopathy, and Profile.



*Figure 13 Edit Profile Page*

**Title:** Edit Profile

**Description:** The Edit Profile screen features a title labeled "Edit Profile" and contains fields for profile information, including the user's name, "Talal Khan," and email, "talalkhan317@gmail.com." The password field is empty, displaying a placeholder with an error message indicating "Password must be at least 6 characters." At the bottom of the screen, there is an action button labeled "Update Profile" for saving changes.



*Figure 14 Article Detail Page*

**Title:** Article Detail

**Description:** The Article Detail screen displays an article titled "Understanding Severe Diabetic Retinopathy," authored by Mohiz and published on June 18, 2024, with an estimated reading time of 5 minutes. This blog provides detailed information where users can read and gain understanding. It is tagged with relevant keywords such as Diabetes, Retinopathy, and Eye Health. The content includes a retinal image with annotations, featuring arrows pointing to specific areas. Highlighted text emphasizes the importance of regular eye exams for the early detection and management of retinopathy. The article explains "No Diabetic Retinopathy" (No DR) as the absence of retinal damage due to diabetes, indicating effective blood sugar management.



## Data Pre-processing

### 5.4.1. Introduction

In the detection of diabetic retinopathy (DR) from fundus images, data preprocessing plays a crucial role. The dataset we worked with presented multiple challenges: inconsistencies in image brightness, significant class imbalance, varying sizes of fundus images, and dark areas surrounding the images. To ensure the reliability and accuracy of our model, we meticulously addressed these issues through a series of preprocessing steps. This section provides a detailed explanation of these steps, demonstrating their importance and effectiveness in improving the quality of the dataset.

### 5.4.2. Image Resizing and Cropping

The fundus images in our dataset varied significantly in size, which posed a challenge for model training. To standardize the images, we resized all images to a uniform size of 224x224 pixels. This was achieved using the Python Imaging Library (PIL), which allows for efficient resizing while preserving the quality of the images.

Additionally, many images had dark areas around the periphery, which did not contribute useful information for DR detection. These areas were cropped to focus on the relevant regions of the fundus. This step was crucial as it reduced noise and ensured that the model received only the most pertinent visual information.

### 5.4.3. Brightness Adjustment

Fundus images in the dataset exhibited varying degrees of brightness, which could negatively impact the model's ability to accurately identify DR. To address this, we applied histogram equalization, a technique that adjusts the contrast of the images by redistributing the intensity values. This method enhances the overall visibility of features in the images, making it easier for the model to detect abnormalities.

Histogram equalization works by flattening the histogram of the image, thereby spreading out the most frequent intensity values. This results in images that are neither too dark nor too bright, providing a more uniform visual representation across the dataset.

### 5.4.4. Application of Gaussian Blur

To further enhance the quality of the fundus images, we applied a Gaussian blur. This technique involves smoothing the images by averaging the pixel values with their neighbors, weighted by a Gaussian function. The purpose of this step was to reduce noise and enhance important features, such as blood vessels and microaneurysms, which are critical for DR detection.

The parameters of the Gaussian blur were carefully selected. We used a sigma value of 10, which determines the extent of the blurring effect. The weighted addition of the blurred image with the original, using the cv2.addWeighted function, helps in highlighting essential features while suppressing irrelevant details.

### 5.4.5. Addressing Class Imbalance

One of the significant challenges in our dataset was the class imbalance. There were many images labeled as "No DR" compared to images with varying severity levels of DR. This imbalance could lead to a biased model that performs well on the majority class but poorly on the minority classes.

To mitigate this issue, we employed data augmentation techniques to artificially increase the number of images in the underrepresented classes. Techniques such as rotation, flipping, and slight adjustments in brightness and contrast were applied to generate new images from existing ones. This approach not only balanced the class distribution but also introduced variability, making the model more robust.

### 5.4.6. Preprocessing Pipeline Implementation

The preprocessing steps were implemented in a Python-based pipeline, ensuring consistency and reproducibility. The pipeline begins by saving the uploaded image and resizing it to 224x224 pixels. Next, the image undergoes cropping to remove dark areas, followed by brightness adjustment through histogram equalization. Finally, a Gaussian blur is applied to enhance critical features.

The following is a high-level overview of the preprocessing steps:

Image Upload and Saving: The uploaded image is saved to a specified directory.

Resizing: The image is resized to 224x224 pixels using the PIL library.

Cropping: Dark areas around the image are cropped to focus on the fundus region.

Brightness Adjustment: Histogram equalization is applied to standardize image brightness.

Gaussian Blur: A Gaussian blur is applied to reduce noise and enhance features.

These preprocessing steps were essential in transforming raw fundus images into a format that is more suitable for model training. By addressing issues related to image size, brightness, and noise, we were able to improve the quality and consistency of the dataset, ultimately enhancing the performance of our DR detection model.

**Conclusion**

Effective data preprocessing is a critical component in the development of a reliable and accurate model for detecting diabetic retinopathy from fundus images. By resizing and cropping images, adjusting brightness, applying Gaussian blur, and addressing class imbalance, we significantly improved the quality of our dataset. These steps ensured that our model received high-quality input data, leading to better detection and classification of DR

## Model Exploration and Comparison

### 5.5.1. Introduction

In our quest to develop an accurate and reliable system for detecting diabetic retinopathy (DR) and classifying its severity from fundus images, we explored several deep learning models. Each model presented unique advantages and challenges, which influenced their performance on our preprocessed dataset. This section provides a detailed analysis of the models we tested, their respective strengths and weaknesses, and the rationale behind selecting EfficientNet as our final model.

### 5.5.2. ResNet

ResNet, or Residual Networks, are known for their ability to train very deep networks by addressing the vanishing gradient problem through residual learning. However, our experiments with ResNet revealed that it was underfitting on our preprocessed data.

**Advantages:**

* ***Residual Learning:*** Allows for training deeper networks without the problem of vanishing gradients, improving performance.
* ***Feature Extraction:*** Effective in extracting features due to its deep architecture.

**Disadvantages:**

* ***Under fitting:*** In our case, ResNet underperformed, likely due to its inability to fully capture the nuances in our dataset.
* ***Complexity:*** Requires careful tuning of hyperparameters and can be computationally expensive.

### 5.5.3. VGG

VGG, known for its simplicity and uniform architecture, was another model we explored. Despite its success in various image classification tasks, VGG exhibited overfitting on our dataset.

**Advantages:**

* ***Simplicity:*** Uniform architecture makes it easy to implement and understand.
* ***Performance:*** Generally performs well on standard image classification tasks.

**Disadvantages:**

* ***Overfitting:*** Tended to overfit our training data, suggesting that it was too complex for our dataset.
* ***Computational Demand:*** Requires significant computational resources for training due to its depth and number of parameters.

### 5.5.4. Xception

Xception, which stands for "Extreme Inception," is based on the Inception architecture but with depthwise separable convolutions. This model also showed signs of overfitting in our experiments.

**Advantages:**

* ***Efficient Convolutions:*** Depthwise separable convolutions reduce computational cost and improve efficiency.
* ***Performance:*** High performance on various image classification benchmarks.

**Disadvantages:**

* ***Overfitting:*** Similar to VGG, Xception overfitted our dataset, indicating it was too complex.
* ***Complexity:*** Implementation and tuning can be more complex compared to simpler models.

### 5.5.5. EfficientNet

EfficientNet emerged as the most effective model for our DR detection task, achieving over 90% accuracy. EfficientNet scales all dimensions of depth, width, and resolution using a compound coefficient, providing a balance between accuracy and efficiency.

**Advantages:**

* ***Scalability:*** EfficientNet scales uniformly, which enhances performance without significantly increasing computational cost.
* ***Accuracy:*** Achieved high accuracy on our preprocessed dataset.
* ***Efficiency:*** More computationally efficient compared to other models with similar performance levels.

**Disadvantages:**

* ***Implementation Complexity:*** Requires careful tuning and understanding of compound scaling.
* ***Training Time:*** Can still be computationally intensive despite its efficiency.

### 5.5.6. Fine-Tuning EfficientNet

After selecting EfficientNet, we further fine-tuned the model using various techniques to enhance its performance:

### 5.5.7. Custom Callback Functions

We implemented custom callback functions to monitor and adjust training processes dynamically. These functions helped in optimizing the model's performance by intervening when necessary.

### 5.5.8. Learning Rate Optimization

A dynamic learning rate adjustment strategy was employed, starting with a higher learning rate and gradually decreasing it as training progressed. This approach allowed the model to converge more efficiently.

### 5.5.9. Regularization Techniques

To prevent overfitting, we applied several regularization techniques:

* ***L1 and L2 Regularization:*** Penalized large weights to maintain simpler models.
* ***Dropout:*** Randomly dropped units during training to prevent co-adaptation and improve generalization.

### 5.5.10. Selection of Loss Optimizers

We experimented with various loss optimizers and found Adamax to be the most effective for our task. Adamax, an extension of Adam, combines the benefits of adaptive learning rate methods with a robust gradient descent approach.

### 5.5.11. EfficientNet B3 Model Structure

After fine-tuning EfficientNet, we specifically chose EfficientNet B3 for its optimal balance between performance and computational efficiency. Post fine-tuning, EfficientNet B3 achieved an impressive accuracy of 95%.

**Model Structure:**

* **Input Shape:** 224x224 pixels, 3 channels (RGB).
* **Base Model:** EfficientNet B3 (pre-trained on ImageNet, excluding the top layer).
* **Additional Layers:**
  + ***Batch Normalization:*** Applied after the base model to stabilize and accelerate training.
  + ***Dense Layer:*** A fully connected layer with 2040 neurons, using L2 regularization for weights and L1 regularization for activities and biases. Activation function is ReLU.
  + ***Dropout Layer:*** A dropout rate of 0.45 to prevent overfitting by randomly setting 45% of the neurons to zero during each update cycle.
  + ***Output Layer:*** A Dense layer with the number of neurons equal to the class count, using a softmax activation function for multi-class classification.

The model was compiled with the Adamax optimizer, a learning rate of 0.001, and categorical cross-entropy as the loss function. Metrics tracked included accuracy.

### 5.5.12. Detailed Explanation:

* ***Image Size and Channels:*** The input images are resized to 224x224 pixels with three color channels (RGB), creating a standardized input shape for the model.
* ***Class Count:*** The number of output classes is determined by the number of distinct classes in the training dataset.
* ***Base Model:*** EfficientNet B3, pre-trained on ImageNet, is used as the backbone of the model. This base model excludes the top layer, allowing us to customize the head for our specific classification task.
* ***Batch Normalization:*** This layer normalizes the output of the base model, stabilizing the learning process and improving training speed and performance.
* ***Dense Layer:*** The dense layer with 2040 neurons employs ReLU activation to introduce non-linearity. Regularization techniques (L1 and L2) are used to prevent overfitting by penalizing large weights and biases.
* ***Dropout Layer:*** This layer helps prevent overfitting by randomly dropping 45% of the neurons during training, ensuring the model does not rely too heavily on any single neuron.
* ***Output Layer:*** The final dense layer uses softmax activation to output a probability distribution over the classes, allowing the model to make a multi-class classification.

The model summary provides a comprehensive overview of the layers, their shapes, and the parameters involved, ensuring transparency and understanding of the architecture.

**Conclusion**

Efficient Net B3, after fine-tuning, demonstrated exceptional performance with a 95% accuracy in detecting and classifying diabetic retinopathy from fundus images. The combination of advanced preprocessing techniques, custom callback functions, dynamic learning rate adjustments, and regularization methods contributed to the robustness and reliability of our model.

## Implementation: Explainable AI with Grad-CAM

### 5.6.1. Introduction

In the field of medical imaging, particularly in detecting diabetic retinopathy (DR), it is crucial not only to achieve high accuracy but also to ensure the model's predictions are interpretable. Explainable AI (XAI) techniques provide insights into the decision-making process of the model, enhancing trust and understanding among users and practitioners. We incorporated Grad-CAM (Gradient-weighted Class Activation Mapping) into our project to visualize and interpret the regions of fundus images that influenced the model's predictions. This section details the implementation of Grad-CAM, discusses its benefits, and explains how we integrated this capability into our system.

### 5.6.2. Grad-CAM Implementation

Grad-CAM is a popular technique for generating visual explanations for deep learning models. It highlights the regions in an input image that are most relevant to the model's prediction by utilizing the gradients of the target class flowing into the final convolutional layer.

**Steps Involved:**

1. **Preprocessing the Image for Grad-CAM:**
   * The input image is resized and preprocessed to match the model's expected input dimensions. This step involves loading the image, resizing it to 224x224 pixels, and converting it into an array suitable for the model.
2. **Model Prediction and Gradient Calculation:**
   * The model predicts the class of the input image. The class index corresponding to the highest predicted probability is identified. Gradients of the predicted class score with respect to the output feature maps of the last convolutional layer are computed.
3. **Generating the Heatmap:**
   * The computed gradients are pooled and weighted. A heatmap is generated by averaging the weighted feature maps along the depth axis. The heatmap is normalized and resized to match the dimensions of the original image.
4. **Overlaying the Heatmap on the Original Image:**
   * The heatmap is superimposed on the original image using a colormap, highlighting the regions that influenced the model's prediction. This overlay provides a visual explanation of the model's decision-making process.

### Benefits of Grad-CAM

1. **Interpretability:** Grad-CAM provides a visual explanation of the model's predictions, helping medical professionals understand which regions of the fundus images influenced the diagnosis.
2. **Trust:** By making the model's decision-making process transparent, Grad-CAM enhances the trustworthiness of the AI system.
3. **Debugging:** Visual explanations help in identifying potential issues or biases in the model, facilitating debugging and improvement.
4. **Compliance:** In medical applications, explainability is crucial for regulatory compliance and ethical considerations.

### Additional Class for Random Images

To further validate and ensure our model's robustness, we added an additional class trained on random images from the internet. This step ensures that the model can distinguish & differentiate between relevant medical images and unrelated irrelevant ones, reducing the possibility of false positives.

**Steps Involved:**

1. **Data Collection:** We gathered a diverse & various set of random images from the internet.
2. **Training:** The model was trained again with this additional class, allowing it to learn the distinguishing traits of non-medical images.
3. **Evaluation:** The model's performance was again evaluated to ensure that it correctly identifies and classifies random images as a separate invalid class.

### Integration with API

To make the Grad-CAM functionality accessible and easy to get, we integrated it into our API system. The endpoint helps provides a proper visual explanation for a given input image, allowing users to upload an image and get a Grad-CAM image overlayed that highlights and shows the critical regions used by the model to make its prediction and evaluation.

### Benefits of Our Approach

1. **Enhanced Interpretability:** By incorporating and adding Grad-CAM, we provide users with visual insights into how the model arrives and comes at its predictions, fostering greater understanding trust & reliance.
2. **Robustness:** Adding a class for random images makes sure that the model can detect differentiate between relevant medical images and irrelevant ones, reducing false positives drastically.
3. **Scalability:** Our deployment on Azure cloud using Docker allows us for easy scaling and maintenance, making sure that the model can handle a high volume of requests efficiently.
4. **Security:** Hosting the model on a secure server reduces the risk of unauthorized access and protects sensitive data.
5. **Efficiency:** The mobile app leverages API calls to interact with the Flask API i.e deployed on Docker. To ensure that the computationally intensive tasks are handled on the server side, maintaining the app's performance.

**Conclusion**

Incorporating and adding Grad-CAM for explainable AI has radically boosted the interpretability and trustworthiness of our DR detection model. By providing visual explanations, we enable medical professionals to understand the model's predictions better, fostering confidence in the AI system. Additionally, adding an extra class for random images ensures the model's robustness and reduces false positives. This comprehensive approach, combined with our deployment strategy on Azure using Docker, makes our solution both powerful and reliable.

# Testing and Evaluation

## Manual Testing

### System Testing

Once the system has been successfully developed, testing has to be performed to ensure that the system working as intended. This is also to check that the system meets the requirements stated earlier. Besides that, system testing will help in finding the errors that may be hidden from the user. There are few types of testing which includes the unit testing, functional testing and integration testing. The testing must be completed before it is being deploy for user to use.

**Objective:** Ensure the system works as intended and meets the requirements.

### Unit Testing:

* + Verify individual units (like login, registration, edit profile, etc.) work correctly.

**Use Case: Registration**

**Objective:** Ensure the registration form is working correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user registration with valid details | Username: Talal Khan, Password:123, Email: talalkahn317@gmail.com | Successfully register and redirect to the login page. | Pass |

**Use Case: Login**

**Objective:** Ensure the login form is working correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user login after clicking on the ‘Login’ button with correct input data | Username: Talal Khan, Password: 123 | Successfully log into the main page of the system. | Pass |

**Use Case: Edit Profile**

**Objective:** Ensure the edit profile form is working correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user can edit their profile with valid data | Name: Taha Malik,  Email: tahamalik567@gmail.com | Profile updated successfully and a confirmation message is shown. | Pass |

**Use Case: Upload Image**

**Objective:** Ensure the image upload functionality is working correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user can upload a valid image | File: retina.jpg | Image uploaded successfully and a confirmation message is shown. | Pass |

**Use Case: View Articles**

**Objective:** Ensure the user can view articles correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user can view the list of articles | - | List of articles is displayed correctly. | Pass |

**Use Case: View Result**

**Objective:** Ensure the user can view their diagnosis result appropriatel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user can view their diagnosis result | - | Diagnosis result is displayed correctly. | Pass |

**Use Case: Log Out**

**Objective:** Ensure the logout functionality is working correctly and precisely.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Verify user can log out successfully | - | User is logged out and redirected to the login page. | Pass |

### Functional Testing:

* + Test different various function to ensure they meet the specifications.

**Use Case: Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Login as a user with valid credentials | Username: user1, Password: pass123 | Main page for the user is loaded with the correct navigation bar. | Pass |

**Use Case: Registration**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Register a new user with valid details | Username: newUser,  Password: newPass123, Email: new@gmail.com | Registration successful, user redirected to login page. | Pass |

**Use Case: Edit Profile**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Edit profile with valid data | Name: newUser,  Email: newUser @gmail.com | Profile updated successfully, confirmation message shown. | Pass |

**Use Case: Upload Image**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upload a valid image file | File: retina.jpg | Image uploaded successfully, confirmation message shown. | Pass |

**Use Case: View Articles**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | View articles | - | List of articles is displayed correctly. | Pass |

**Use Case: View Result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | View diagnosis result | - | Diagnosis result is displayed correctly. | Pass |

**Use Case: Log Out**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Log out from the system | - | User logged out successfully, redirected to login page. | Pass |

### Integration Testing:

* + Verify combined various parts of the system function together as expected.

**Scenario: User Workflow**

*Table 13 user workflow*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Register, Login, and Edit Profile | Username: newUser,  Password: newPass123,  Email: [new@gmail.com](mailto:new@gmail.com);  Login:  Username: newUser,  Password: newPass123;  Edit Profile:  Name:talal,  Email: talal@gmail.com | Successfully register, login, and update profile. | Pass |
| 2 | Upload Image and View Result | File: retina.jpg | Successfully upload image and view diagnosis result. | Pass |
| 3 | View Articles and Log Out | - | Successfully view articles and log out. | Pass |

# Conclusion and Future Work

## Conclusion

The development of an innovative cutting-edge machine learning-based system for automated retinopathy detection/findings represents significant advancement in ocular healthcare. By leveraging advanced machine learning algorithms/systems and Explainable AI (XAI) techniques, this project has successfully addressed the crucial need for early detection of retinopathy, a severe condition occurring among hypertensive and diabetic patients. Our robust and accurate model processes retinal images to classify different stages and levels of retinopathy, thereby providing vital support to healthcare professionals in making informed decisions timely. The integration of XAI techniques such as trainable attention, CAM, Guided Grad-CAM, Grad-CAM++, and Multiple Instance Learning boosts the interpretability of the model, fostering exceptional trust and transparency in AI-driven healthcare solutions. Through comprehensive dataset collection, rigorous model training/testing, and thorough evaluation, we have established a user-friendly interface that facilitates seamless and flawless interaction with the system. This project not only underscores the aptitude of AI in revolutionizing medical diagnostics but also highlights the importance of transparency and interpretability in AI applications.

## Future Work

While the current project has made significant and exceptional strides in the early detection and interpretability of retinopathy diagnosis, there are several avenues for future work and improvement to further enhance and expand its proficiencies:

**Expansion of Dataset:** Increasing the size and diversity of the dataset with retinal images from various populations and medical settings will improve the model's robustness and generalization. Collaborating with international medical organizations can help in acquiring a more comprehensive dataset.

**Integration with Electronic Health Records (EHRs):** Integrating the system with EHRs can provide a better holistic view of a patient's health, allowing for personalized diagnosis and treatment plans. This integration can also enable longitudinal studies to monitor disease progression.

**Real-time Processing and Deployment:** Enhancing the system such that it support real-time image processing and deploying it in clinical settings and places will be crucial for practical applications. This involves optimizing the model for speed performance and ensuring it meets regulatory standards for medical devices.

**Multi-modal Data Integration:** Incorporating additional data modalities, such as patient history, genetic information, and other diagnostic images, can improve the accuracy and comprehensiveness of the diagnosis.

**Patient and Clinician Feedback Loop:** Establishing a feedback loop where patients and clinicians can offer input on the system's performance and usability will help in continuously refining and enhancing the interface and functionality.

**Continuous Learning and Adaptation:** Implementing mechanisms for continuous and non-stop learning where the model can be updated with new data and feedback will ensure that the system remains up to date with current latest medical knowledge and practices.

**Exploration of Other Medical Applications:** Extending the framework to detect and discover other ocular diseases or even other types of medical conditions using similar techniques can broaden the impact of this technology.

**Regulatory Approval and Ethical Considerations:** Working towards obtaining regulatory support for clinical use and addressing ethical deliberations related to AI in healthcare will be essential for the widespread adoption & acceptance foe this technology.

By pursuing these future directions, moreover we can further enhance the efficacy, reliability, and adoption of AI-driven solutions in healthcare, eventually contributing to better patient outcomes and progressing the field of medical diagnostics.

**References**

1. Malhi, A., Grewal, R., & Pannu, H. S. (2023). Detection and diabetic retinopathy grading using digital retinal images. International Journal of Intelligent Robotics and Applications, 7(2), 426–458. https://doi.org/10.1007/s41315-022-00269-5
2. Butt, M. M., Iskandar, D., Abdelhamid, S., Latif, G., & Alghazo, R. (2022). Diabetic Retinopathy Detection from Fundus Images of the Eye Using Hybrid Deep Learning Features. Diagnostics, 12(7), 1607. https://doi.org/10.3390/diagnostics12071607
3. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, Oct. 2019, doi: 10.1007/s11263-019-01228-7.
4. “‘Home,’” *Docker Documentation*, May 21, 2024. https://docs.docker.com/
5. “App Service — Build and host web apps | Microsoft Azure.” https://azure.microsoft.com/en-us/products/app-service/
6. “Students - GitHub Education,” *GitHub*, 2024. https://github.com/education/students
7. “Welcome to Flask — Flask Documentation (2.0.x).” https://flask.palletsprojects.com/en/2.0.x/
8. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *arXiv.org*, Dec. 10, 2015. https://arxiv.org/abs/1512.03385
9. K. Simonyan and A. Zisserman, “Very deep convolutional networks for Large-Scale image recognition,” *arXiv.org*, Sep. 04, 2014. https://arxiv.org/abs/1409.1556
10. F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” *arXiv.org*, Oct. 07, 2016. https://arxiv.org/abs/1610.02357
11. M. Tan and Q. Le V., “EfficientNet: Rethinking model scaling for convolutional neural networks,” *arXiv.org*, May 28, 2019. https://arxiv.org/abs/1905.11946
12. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv.org*, Dec. 22, 2014. <https://arxiv.org/abs/1412.6980>
13. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. Retrieved from<https://arxiv.org/abs/1610.02391>
14. Docker. Retrieved from https://docs.docker.com/
15. Microsoft Azure. Retrieved from<https://azure.microsoft.com/en-us/services/app-service/>
16. GitHub Education. Retrieved from<https://education.github.com/pack>
17. Flask. Retrieved from https://flask.palletsprojects.com/en/2.0.x/