A PROJECT REPORT ON

Recognition of Ocular Diseases using Deep Learning Techniques

SUBMITTED TO THE SAVITRIBAI PHULE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (Computer Engineering)

SUBMITTED BY

Sameer Memon	Exam No: B190054383
Sejal Pekam	Exam No: B190054432
Soham Naik	Exam No: B190054398
Kirti Palve	Exam No: B190054409

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune - 411043

SAVITRIBAI PHULE PUNE UNIVERSITY
2022-2023

CERTIFICATE

This is to certify that the project report entitled

Recognition of Ocular Diseases using Deep Learning Techniques

Submitted by

Sameer Memon	Exam No: B190054383
Sejal Pekam	Exam No: B190054432
Soham Naik	Exam No: B190054398
Kirti Palve	Exam No: B190054409

is a bonafide student of this institute and the work has been carried out by him/her under the supervision of **Dr. B. A. Sonkamble** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of **Bachelor of Engineering** (Computer Engineering).

Dr. B. A. Sonkamble
Internal Guide
Dept. of Computer Engg.

Dr. G.V. Kale
Head
Dept. of Computer Engg.

Dr. S. T. Gandhe
Principal
Pune Institute of Computer Technology

Place: Pune
Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the project report on ***'Recognition of Ocular Diseases using Deep Learning Techniques'****.*

We would like to take this opportunity to thank ***Dr. B. A. Sonkamble*** *giving me all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Dr. G. V. Kale****, Head of Computer Engineering Department, PICT for his indispensable support, suggestions.*

Sameer Memon
Sejal Pekam
Soham Naik
Kirti Palve
(B.E. Computer Engg.)

ABSTRACT

Ocular disease refers to the deteriorating performance and working of the human eye. Globally, fundus disorders are the primary cause of blindness in humans. The timely detection of ocular disease is crucial for prevention of blindness caused by diseases like diabetes, glaucoma, cataract, age-related macular degeneration (AMD). Therefore, an automated ocular disease detection system with computer-aided tools is necessary to detect various eye disorders using fundus pictures. It is possible to create such a system given the advancements in the field of deep learning and certain advanced image classification techniques. A structured database,Ocular Disease Intelligent Recognition (ODIR), comprising data of 5,000 patients with age, color fundus photographs from left and right eyes, and doctors' diagnostic keywords from doctors is used for the same. It contains eight different classes of the fundus. These classes represent different ocular diseases.
Hence, the techniques used in this project aim to provide an efficient solution to detect the presence of multiple eye diseases from the fundus images.

Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Overview

This project focuses on identifying the need for accurate and efficient detection of ocular diseases, such as diabetic retinopathy, age-related macular degeneration, glaucoma, etc. and highlighting the importance of early detection and intervention to prevent vision loss and improve patient outcomes.A diverse and representative dataset of ocular images, including both healthy and diseased eyes is gathered from the ODIR-5k dataset which is a benchmark dataset. Perform preprocessing steps on the dataset, such as resizing, normalization, and augmentation.Split the dataset into training, validation, and testing sets to evaluate model performance. Design a CNN architecture suitable for ocular disease detection. In this case, VGG-16 and Resnet-50 has been used. Use convolutional layers to capture spatial features in the images. Employ pooling layers to reduce dimensionality and extract dominant features. Incorporate fully connected layers for classification. Initialize the CNN model with random weights. Utilize the training set to optimize the model's parameters through backpropagation and gradient descent. Experiment with hyperparameter tuning, such as learning rate, batch size, and regularization techniques. Monitor training progress and evaluate performance on the validation set.Assess the trained model's performance on the testing set to obtain unbiased results. Measure key metrics like accuracy, precision, recall, and F1 score.Analyze the model's confusion matrix to understand specific disease detection strengths and weaknesses. Explore techniques like transfer learning to leverage pre-trained models on large-scale image datasets. Iterate on the model architecture and hyperparameters to improve overall performance.Package the trained model into a deployable format. Build a user-friendly interface to accept ocular images for disease detection. Integrate the model into an application or system that can be accessed by healthcare professionals or patients. Ensure scalability, reliability, and privacy considerations.

1.2 Motivation

Ocular disease refers to any condition or disorder that interferes with the capacity of the eye to operate correctly. It can be challenging for doctors to identify eye disorders early enough using fundus pictures. Diagnosing ocular illnesses manually is time-consuming and complicated. Therefore, an automated ocular disease detection system with computer-aided tools is necessary to detect various eye disorders using fundus pictures. It is possible to create such a system given the advancements in the field of deep learning and certain advanced image classification techniques. We plan to use these advanced image classification techniques to classify the presence of certain diseases by creating a machine learning pipieline on the dataset containing images of the fundus of the human eye.

1.3 Problem Definition and Objectives

ODIR- To develop an automated pipieline for the recognition of ocular diseases such as Diabeteic Retinopathy, Glaucoma, Cataract and Hypertension by implementing deep learning algorithms on the dataset containing opthalmic images of the human eye fundus.

1.4 Project Scope & Limitations

Scope:

- Design and develop a user-friendly website for detecting ocular disease.
- Include features such as upload image, detect if correct image uploaded, run processing to check if ocular disease is found and display results.
- Support integration with third-party services like camera apps and image uploaders.
- Develop responsive design for optimal user experience across devices.
- Provide basic reporting and analytics capabilities for administrators.

Limitations:

- Project limitations define the constraints and restrictions that may impact the project's execution and deliverables.
- The project will focus on the development of the website only and will not include the development of native mobile applications.
- The website will be developed for a single language.
- The project budget and timeline may not allow for extensive customization or integration with complex enterprise systems.
- The website will be developed for standard web browsers and may not be fully optimized for older or less commonly used browsers.

- Dataset on which model is trained is specific to one geographical region i.e. China.
- The project will not include extensive search engine optimization (SEO) or marketing campaigns; it will focus primarily on the website's technical development.

1.5 Methodologies of Problem solving

Agile methodologies promote adaptive and iterative development, allowing teams to respond to change and deliver software in shorter cycles. Some popular Agile methodologies include: Scrum: It divides work into short, time-boxed iterations called sprints, with frequent feedback and flexibility. Kanban: It visualizes the workflow and focuses on continuous flow, minimizing work in progress and optimizing throughput. Extreme Programming (XP): It emphasizes customer involvement, frequent releases, continuous testing, and pair programming.

CHAPTER 2

LITERATURE SURVEY

2.1 A Benchmark of Ocular Disease Intelligent Recognition: One Shot for Multi-disease Detection, 2021

Ocular Disease Detection is critical to prevent blindness caused by diseases like Galucoma, Diabetetes, Age related diseases etc. Ocular refers to diseases that affect the working and operation of the eye. The detection is carried out by analysing fundus images. To overcome problems with existing datasets a new dataset is introduced having 10,000 images from 5,000 patients with multiple modals for multiple diseases.The dataset is skewed in favour of certain diseases which occur more often as the availability of data for such diseases is more.

2.2 Deep Learning for Ocular Disease Recognition: An Inner-Class Balance, 2022

Previously, researchers have carried out the detection of Ocular diseases using various computer vision techniques using deep learning and neural network. This is done to ensure and lower the risk of blindness caused by ocular diseases. Ocular diseases refer to disorders related to the eye.However, the dataset of fundus images used was skewed.To overcome the skewness in data multiclass classification is converted to binary classification with balancing in the dataset in this particular paper. Researchers can further also use GANs for the creation of datasets to overcome the skewness. This helps provide a larger variety of unbiased and normalised dataset and also aids in preventing the problem of overfitting.

2.3 Rethinking the Inception Architecture for Computer Vision, 2016

A common technique such as use of multiple layers in the neural network resulted in overfitting of models. Thus, the use of Inception V1 was looked upon which uses multiple filters on the same level using a wider architecture rather than deeper. The aim is to increase the width and improve the accuracy while keeping the computing resources constant.Inception focuses on parallel processing. It extracts various feature maps concurrently. Deep neural networks are computationally expensive. To make it cheaper, the authors limit the number of input channels by adding an extra 1x1 convolution before the 3x3 and 5x5 convolutions. Reducing the dimensions too much can cause a loss of information. On the other hand, shallow networks are not as efficient in terms of computational accuracy and cost. The learning rate is also affected in the process.

2.4 Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs, 2016

In this study, deep learning was used to train an algorithm to detect referable diabetic retinopathy and assess the performance of the algorithm on validation sets.Further research is necessary to determine the feasibility of applying this algorithm in the clinical setting and to determine whether use of the algorithm could lead to improved care and outcomes compared with current ophthalmologic assessment.

2.5 Ophthalmic Disease Detection via Deep Learning With a Novel Mixture Loss Function, 2021

The accuracy is further improved by incorporation of loss functions in the neural network model. Authors presented a mixture of those two losses in deep neural network model to improve the recognition performance of classifier for biomedical data. The majority of DL-based methods are interested to a unique ocular pathology. However, the clinical context requires detecting several eventual diseases in the same screening, which correspond to a real challenge. However, due to loss function, it becomes sensitive to outliers.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

In this section, we study the Assumptions and Dependencies of our project, user classes and characteristics along with assumptions and dependencies of our project.

3.2 Functional Requirements

The following are the functional requirements for the proposed system:

3.2.1 Capture the image of the fundus

Medical experts upload the image of the fundus of the patient.Prior to this, the image has to be captured using a special method and appropriate camera and equipment under an expert supervision.

3.2.2 Data pre-processing of the available dataset

The dataset is skewed towards cataract more than the other diseases. This may create a bias while training the model and hence the accuracy of other disease labels may be less.The images in the data are very big and have different image resolutions. Most images have sizes of around 2976x2976 or 2592x1728 pixels. Some of the images may be of low quality due to factors like lens dust or haziness.

3.2.3 Model training

The model is trained on the dataset which contains around 8000 training images. This is done using algorithms like convolutional neural networks, vgg-16, inception v3, resnet 50.

3.2.4 Uploading the image on the web application

The captured image of the fundus is uploaded on the web application. The application verifies the admissibility of the image.

3.2.5 Receive an elaborate final report

The trained model performs analysis on the received fundus image to classify the image into one of the eight classes.

3.3 External Interface Requirements

3.3.1 User Interfaces

ODIR Client: To create a web application for the user to upload image of the fundus. Based on the quality of image, the application may accept or display error. The image is then sent to the trained Machine Learning model and corresponding result is displayed to the user.

3.3.2 Software Interfaces

1. Operating System: Anyone among MAC OSX, Linux or Windows
2. Jupyter Notebook
3. Dataset: ODIR
4. Programming Language: Python

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

1. User satisfaction:

   It evaluates if the application and services offered meet or exceed the customer's expectations. The system need to be easy to use.

2. Average response time:

   The time it takes the Application Server to respond to a user's request and return the results is known as the average response time. The model should evaluate within a reasonable time.

3. Accuracy:

   Accuracy depends on how well the system detects a potential ocular disease based on the provided fundus images .

4. Application Availability:

   The degree to which an application is operational, functional, and usable for completing or satisfying a user's or business's requirements is referred to as its availability.

3.4.2 Safety Requirements and Security Requirements

1. User Security:

   User information related to the result must be kept confidential.

3.4.3 Software Quality Attributes

1. Correctness

   The compliance of programme code with specifications and the independence of the software system's actual application are both indicators of a software system's correctness.

2. Reliability

   The reliability of a software system is defined as the likelihood that it will perform a function (given by certain requirements) for a specified number of input trials under a specified number of input conditions in a specified time period (assuming that hardware and input are free of errors).

3. Learnability

   The user interface design and the readability and simplicity of the user instructions (tutorial or user manual) are both factors that affect how easy it is to learn a software system.

4. Robustness

   The impact of operational mistakes, incorrect input data, and hardware failures is reduced by robustness.

3.5 System Requirements

3.5.1 Database Requirements

We have the following data that needs to be stored:

1. Image ID.
2. Patient name.
3. Patient sex.
4. Left fundus.
5. Right fundus.
6. Left diagnostic keywords.
7. Right diagnostic keywords.
8. Various corresponding classes.

3.5.2 Software Requirements (Platform Choice)

1. Latest version of browser like Google Chrome, Mozilla firefox etc.
2. Python
3. Framework: Tensorflow
4. Operating System: Windows, Mac OS, Linux etc.
5. Algorithm: CNN, VGG, RESNET,Inception V3

3.5.3 Hardware Requirements

1. Processor (i5 or higher): A fast and efficient processor is needed to train the model.
2. RAM (8GB minimum): Helps to boosts the performance of system.
3. GPU .

3.6 Analysis Models: SDLC Model to be applied

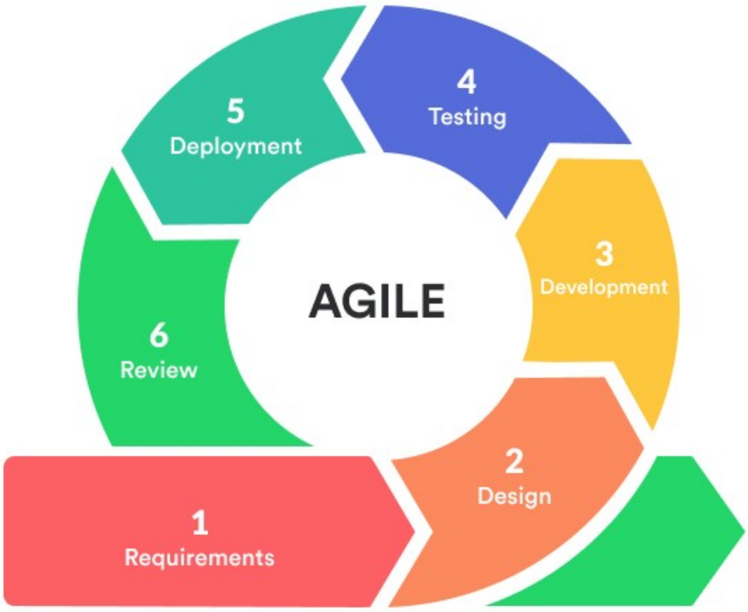

Figure 3.1: Agile Model for SDLC

CHAPTER 4

SYSTEM DESIGN

4.1 Architecture diagram

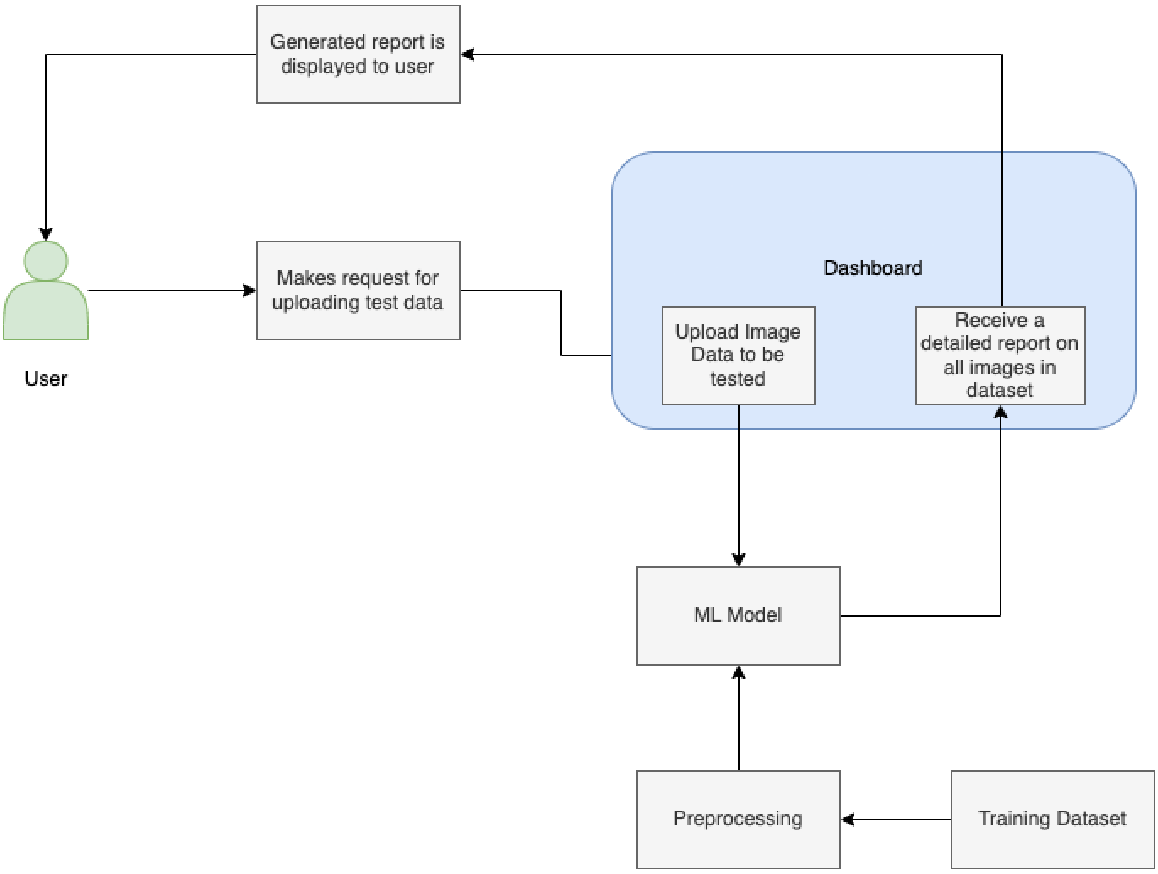

Figure 4.1: Architecture diagram

4.2 Data Flow Diagrams

4.2.1 Data Flow Diagram Level-0

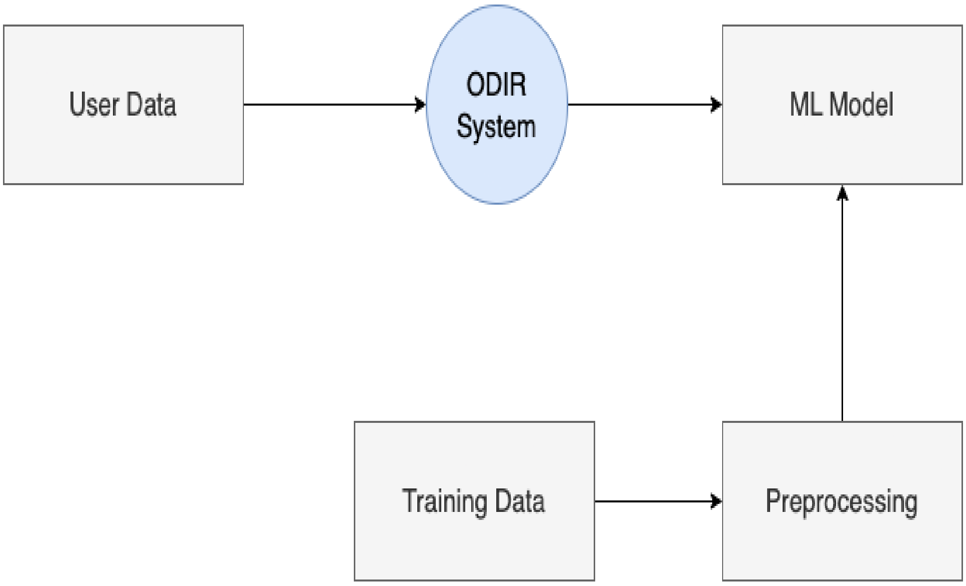

Figure 4.2: DFD level-0

4.2.2 Data Flow Diagram Level-1

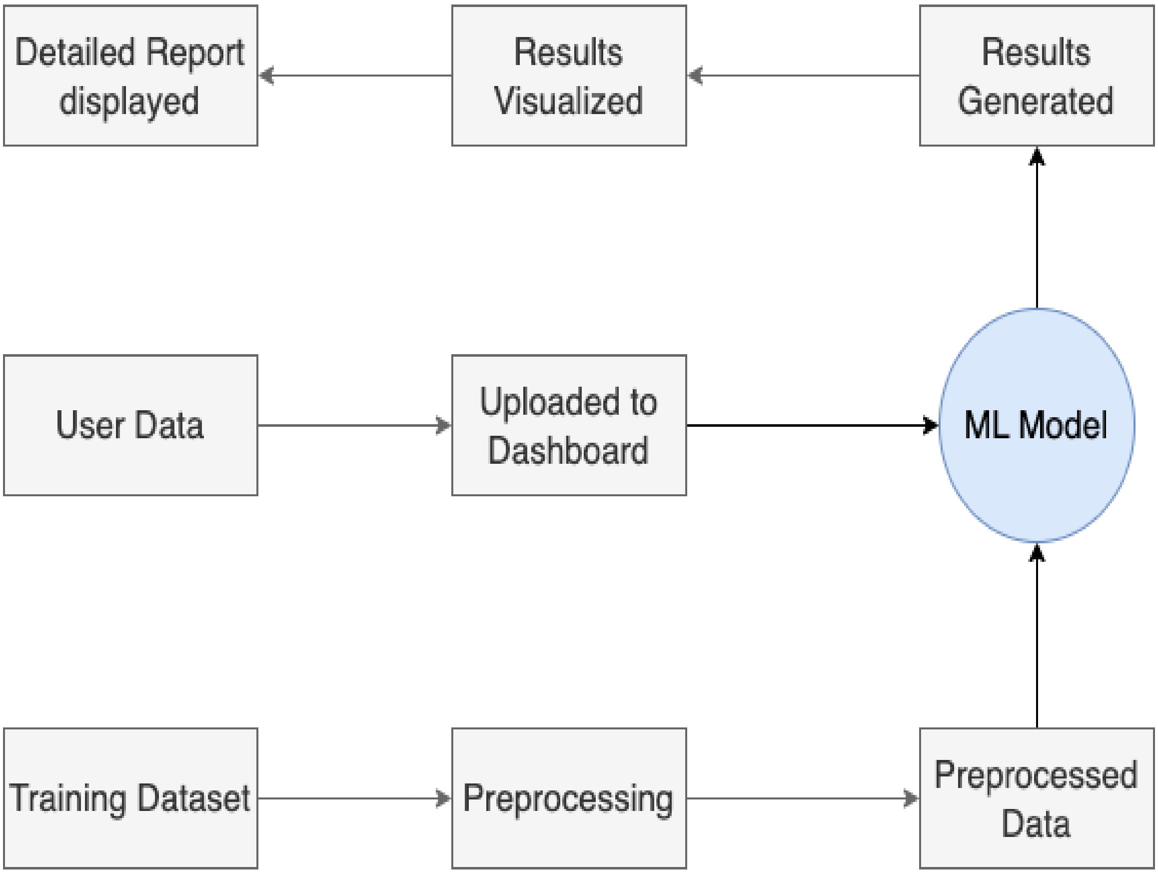

Figure 4.3: DFD level-1

4.3 Use Case Diagrams

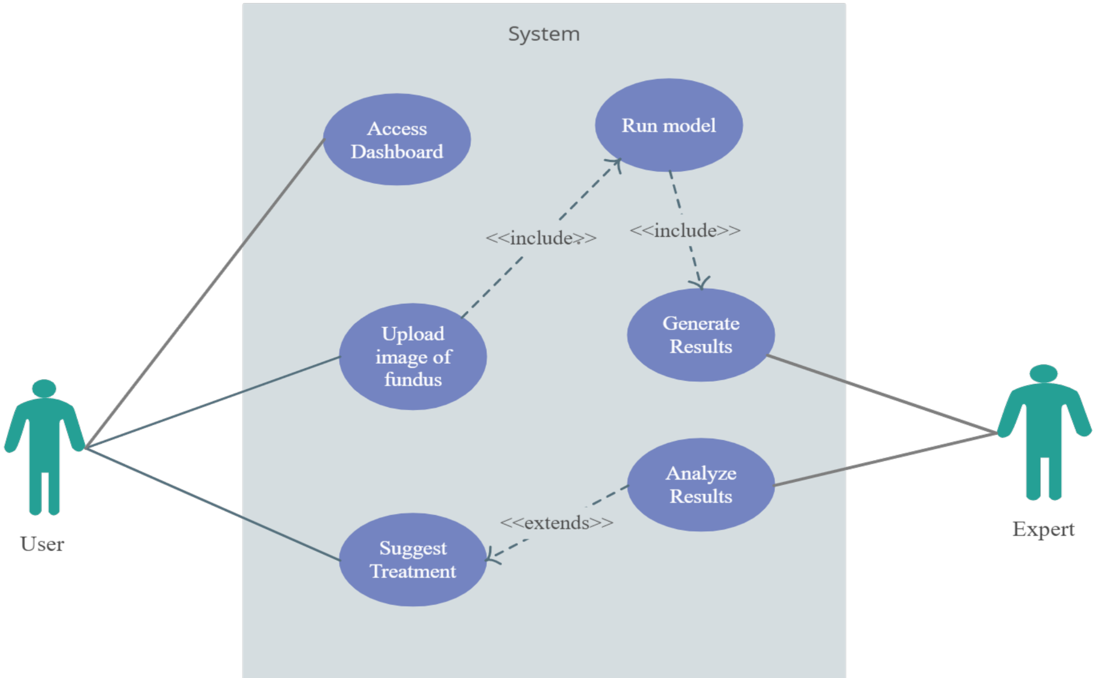

Figure 4.4: Use Case Diagram

4.4 Class diagram

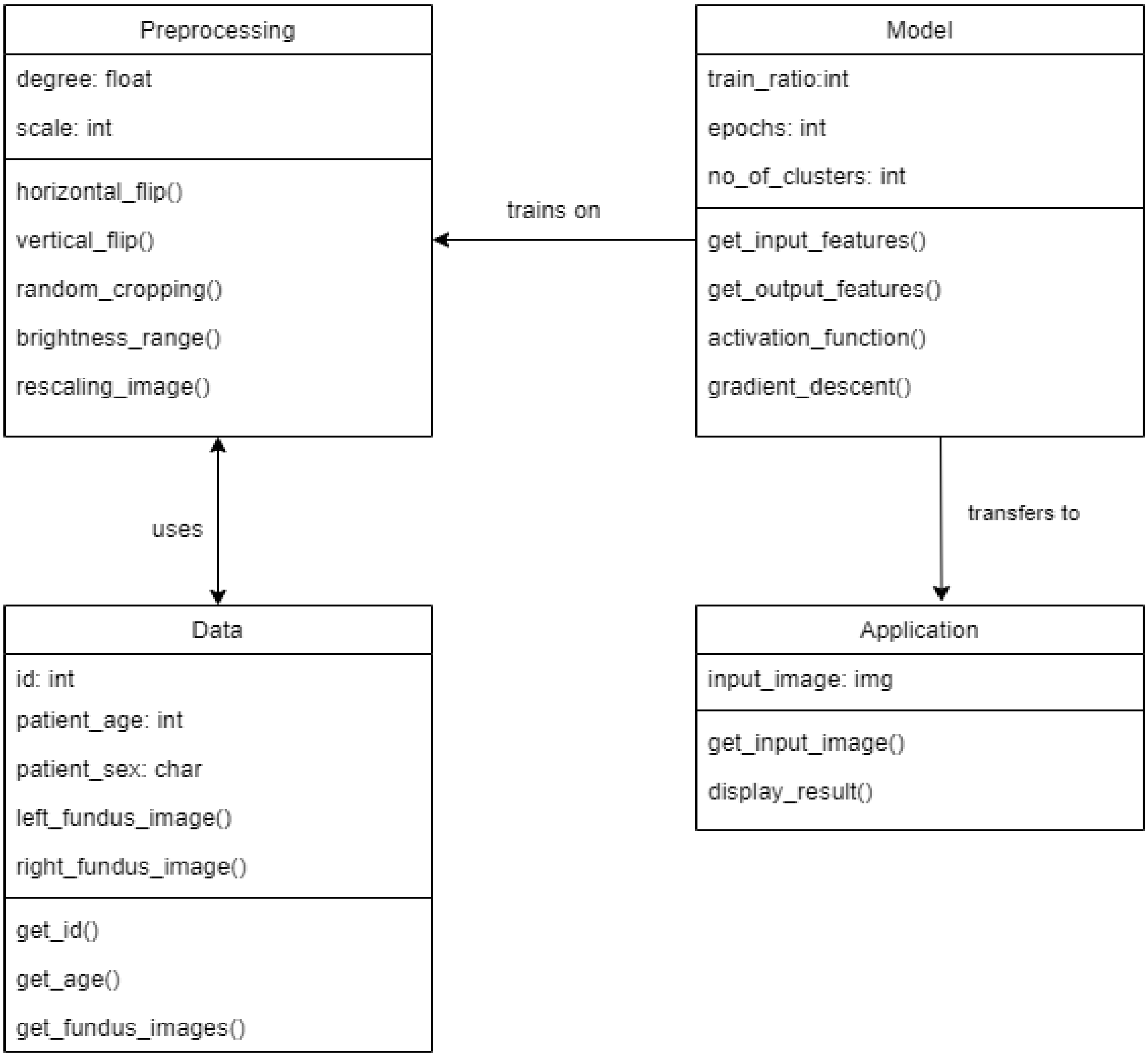

Figure 4.5: Class diagram

4.5 Entity Relationship Diagram

Here, we have shown how two different entities will interact using the system architecture.

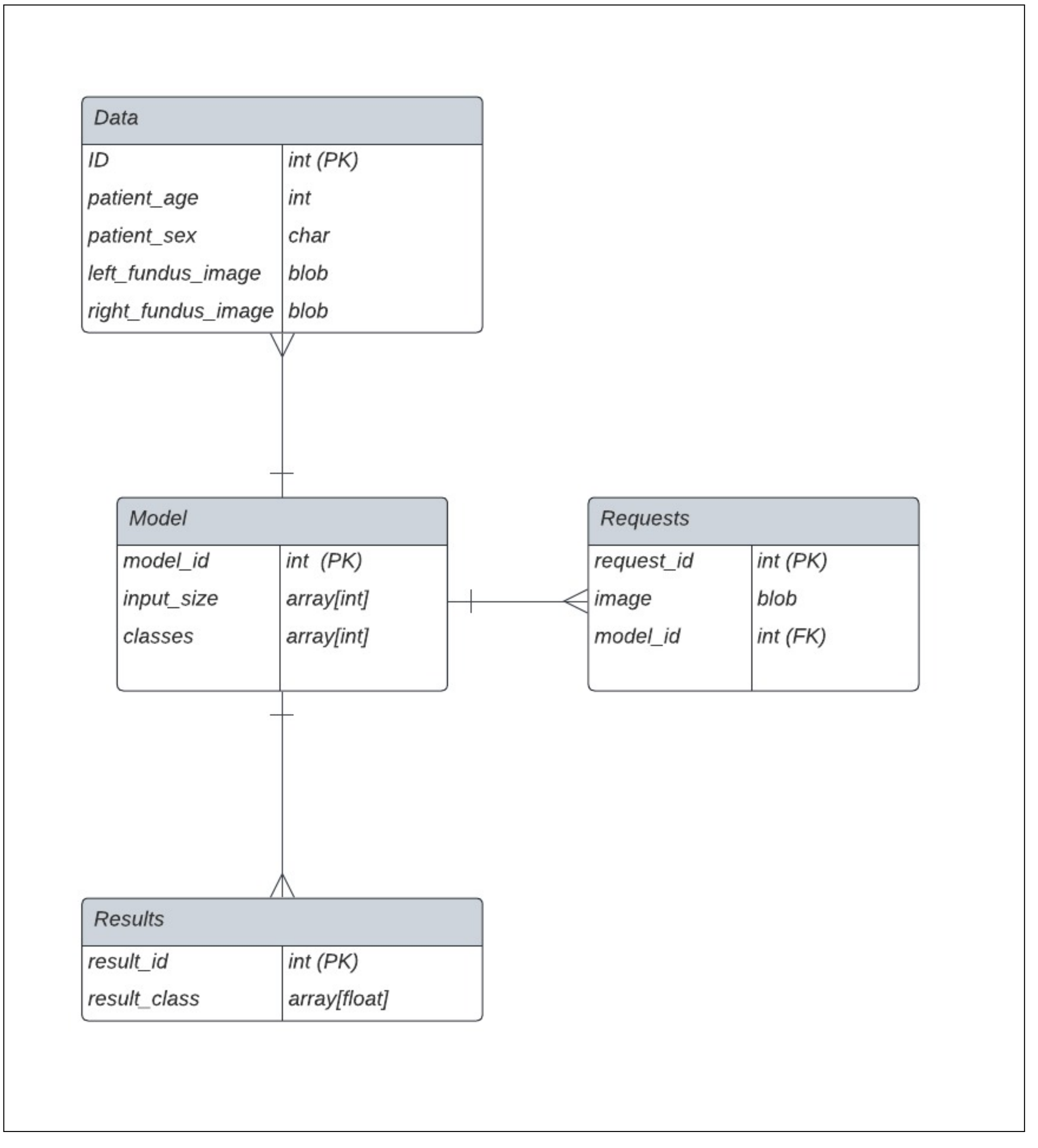

Figure 4.6: Entity Relationship diagram

4.6 Sequence Diagram

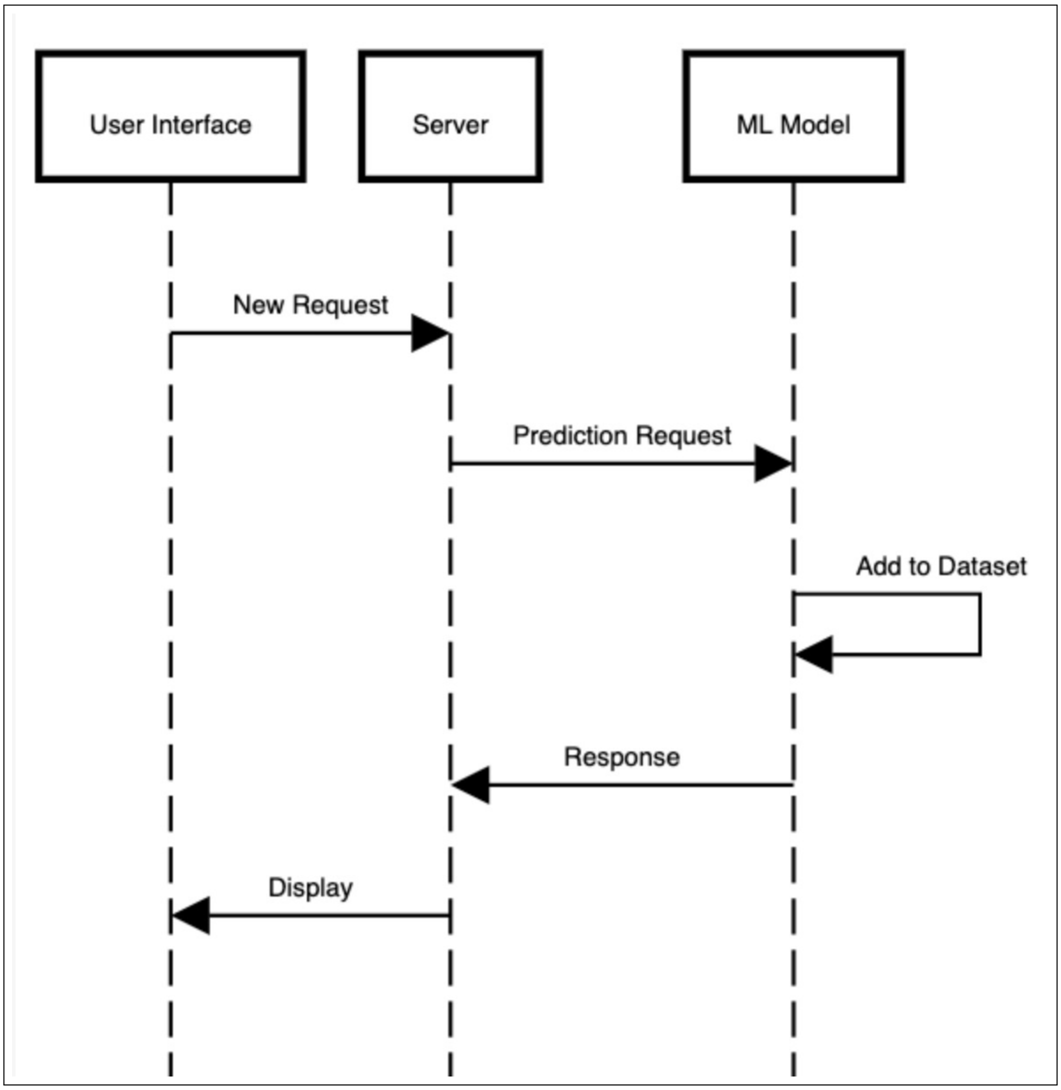

Figure 4.7: Sequence diagram

CHAPTER 5
PROJECT PLAN

5.1 Project Estimate

5.1.1 Reconciled Estimates

- Cost Estimate: The software used for the application will incur no cost as it is all released under free open-source licenses.
- Time Estimate: 8-10 months

5.1.2 Project Resources

- Number of Human Resources: 4
- Proficiency in various programming languages, machine learning, and deep learning concepts and algorithms.
- Clients for whom this model is developed are doctors, researchers, scientists, clinics or anyone who has access to fundus images.
- In any project, stakeholders are individuals or groups with a vested interest in the outcome of the project. For this project, the stakeholders will include both the clients who will be using the application and the project team responsible for developing it.

5.2 Risk Management

Risk management in software engineering projects is a critical process that helps identify, assess, and mitigate potential risks that can impact the success of a project. By proactively addressing risks, project teams can minimize their negative effects and improve the overall project outcomes.

5.2.1 Risk Identification

- Technical risk: There is a risk of encountering compatibility issues with different mobile devices and operating systems.
- Scope creep: There is a possibility of the client requesting additional features or changes that could impact the project timeline and resources.
- Resource constraints: There is limited access to additional dataset pertaining to local patients required for the project.
- Unrealistic deadlines: The project has a tight deadline, and there is a risk of not being able to meet it due to unexpected challenges or delays like reduced scope for verification of algorithms due to limited dataset.

5.2.2 Risk Analysis

The identified risks are analysed based on their probability and impact. They rank the risks as high, medium, or low based on the severity of their potential consequences.

- Technical risk: High probability and high impact, as it can lead to significant development delays and customer dissatisfaction.
- Scope creep: Medium probability and medium impact, as it may require additional development efforts and impact the project timeline to some extent.
- Resource constraints: Low probability and medium impact, as it may require finding alternative solutions or acquiring additional resources to mitigate the risk.
- Unrealistic deadlines: High probability and high impact, as failure to meet the deadline may result in penalties or loss of client trust.

5.2.3 Overview of Risk Mitigation, Monitoring, Management

Risk Mitigation

- Technical risk: An analysis is carried out to decide which ocular disease dataset is the most suitable and appropriate.
- Scope creep: There is regular trial and error process as well as communication to clarify the exact requirements of the process and approximate the target demographic, in this case, medical professionals.
- Resource constraints: The project plan is adjusted according to skills of the various group members as well as resources available.
- Unrealistic deadlines: The team maintains a well-defined schedule, closely monitors progress, and identifies any potential delays early on. They communicate with the client and stakeholders, discussing potential trade-offs and alternative solutions to meet the deadline.

Risk Monitoring: Throughout the project, the team monitors the identified risks, tracks their status, and evaluates the effectiveness of risk mitigation strategies. They conduct regular risk reviews and adjust their approach as needed.
Risk Response:

- Technical risk: The team promptly addresses compatibility issues by analyzing the root cause and implementing necessary fixes or workarounds.
- Scope creep: The team follows the change control process, evaluating the impact of requested changes and adjusting the project plan and resources accordingly.
- Resource constraints: The team explores alternative solutions, such as outsourcing specific tasks or collaborating with external experts, to mitigate resource limitations.
- Unrealistic deadlines

5.3 Project Schedule

5.3.1 Project Task Set

- Requirements Gathering and Analysis
- System Design
- Development
- Testing
- Documentation
- Deployment
- User Training
- Maintenance and Support
- Project Management

5.3.2 Timeline Chart

Table 5.1: Timeline Chart

Deadline	Task	Status	Assigned To
August	Basic redevelopment of code	Completed	All
September	Timeout errors	Completed	All
October	Dataset Verification	Completed	Sejal
November	Model survey	Completed	Soham, Kirti
December	Research multilabel multiclass techniques	Completed	Sameer, Soham
January	Streamlit techniques for front end	Completed	Sejal
February	Review present techniques	Completed	Kirti
March	Deploy backend	Completed	Sameer, Sejal
April	Test model	Completed	Soham

5.4 Team Organization

The project team's organization and reporting mechanisms are described. Regular updates on project progress are provided to the project guide through meetings held once or twice a month.

5.4.1 Team structure

The team structure is established, and roles are defined as follows:

- Kirti Palve: ML Engineer
- Sameer Memon: ML Engineer
- Sejal Pekam: ML Engineer
- Soham Naik: ML Engineer

5.4.2 Management reporting and communication

The team maintains regular communication with the project guide and collaborates in person to improve efficiency. Additionally, chat groups, emails, and online meetings are utilized for communication purposes.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

- User Interface (UI): This module focuses on the design and development of the graphical user interface (GUI) or the user experience (UX) components of the software using Streamlit, which is a python based library used for machine learning engineers.
- Database Management: This module deals with the storage, retrieval, and management of data within the software. It includes tasks such as database design, data modeling, and integration with database systems.
- Business Logic: The business logic module encapsulates the core functionality and rules of the software. It implements the algorithms, workflows, and rules that define how the software operates and processes data.
- Testing and Quality Assurance: This module focuses on ensuring the quality and reliability of the software through various testing techniques, such as unit testing, integration testing, and system testing. It also includes tasks related to bug fixing and performance optimization.
- Deployment and Infrastructure: This module involves the setup and configuration of the software environment, including deployment to production servers, scalability considerations, and monitoring of the software's performance and availability.
- Documentation and User Support: This module deals with creating user documentation, such as user manuals and guides, as well as providing support to end-users, answering their questions, and resolving issues they may encounter while using the software.

6.2 Tools and Technologies Used

- Google colab IDE
- Version control - Git
- Issue Tracking System - Jira
- Testing framework - PyTest for Python
- Collaboration and Communication Tool - Microsoft Teams, Google meet

- Agile Project Management
- Primary programming language used - Python

6.3 Architecture Details

6.3.1 VGG-16

The VGG16 (Visual Geometry Group 16) is a popular convolutional neural network (CNN) architecture used for image classification tasks. It was developed by the Visual Geometry Group at the University of Oxford. Here is a high-level description of the VGG16 algorithm:

1. **Input Layer**: The input to VGG16 is a 224x224 RGB image.
2. **Convolutional Layers**: VGG16 consists of 13 convolutional layers, denoted by "conv" followed by a number. Each convolutional layer uses a small receptive field (3x3) and applies a set of learnable filters to the input. The number of filters increases as we go deeper into the network. These convolutional layers are followed by a Rectified Linear Unit (ReLU) activation function to introduce non-linearity.
3. **Max Pooling Layers**: After each set of convolutional layers, VGG16 includes a max pooling layer denoted by "pool" followed by a number. The max pooling operation reduces the spatial dimensions of the feature maps, making the network more robust to translations in the input image.
4. **Fully Connected Layers**: VGG16 has three fully connected layers, denoted by "fc" followed by a number. These layers take the flattened output from the previous layers and perform classification. Each fully connected layer is followed by a ReLU activation function, except for the last layer.
5. **Dropout**: To prevent overfitting, dropout regularization is applied to the first two fully connected layers. Dropout randomly sets a fraction of the inputs to 0 during training, which helps prevent the network from relying too heavily on specific features.
6. **Output Layer**: The final fully connected layer, denoted as "fc8", produces the output logits. For image classification tasks, this layer typically has the same number of units as the number of classes in the dataset.

7. **Softmax Activation**: A softmax activation function is applied to the logits to obtain normalized class probabilities. This allows us to interpret the output as the model's confidence scores for each class.

8. **Loss Function**: The standard loss function used with VGG16 for image classification is the cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the true labels.

9. **Training**: VGG16 is trained using backpropagation and gradient descent optimization. During training, the model learns the optimal weights for the filters in the convolutional layers and the weights in the fully connected layers.

10. **Inference**: Once trained, VGG16 can be used to make predictions on new unseen images by passing them through the network and obtaining the class probabilities from the softmax layer.

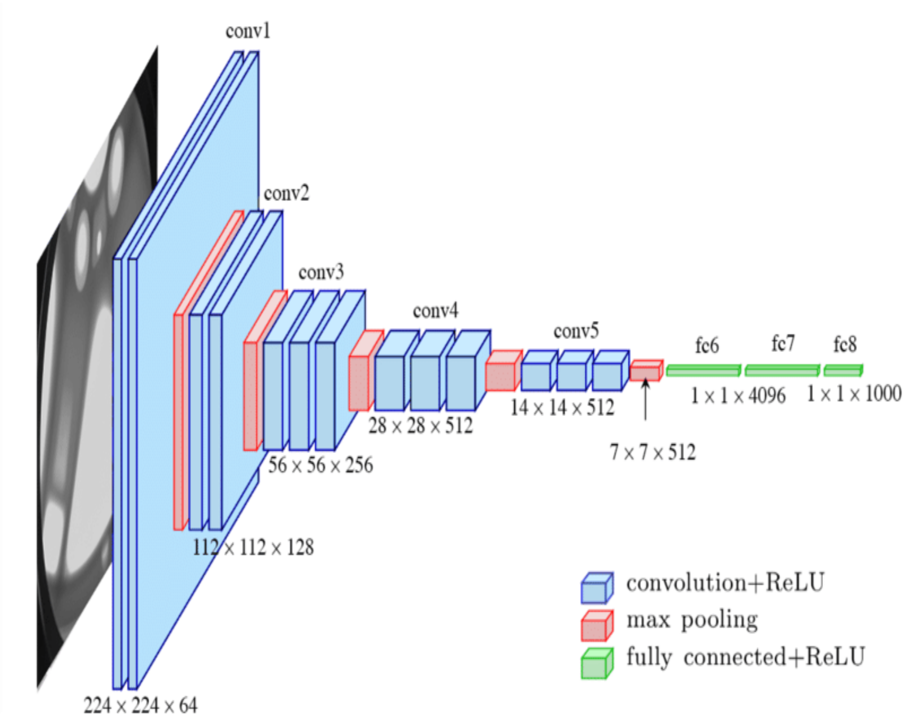

Figure 6.1: VGG-16 Architecture

6.3.2 ResNet50

The ResNet-50 (Residual Network-50) is a deep convolutional neural network architecture that was introduced by Microsoft Research in 2015. It is known for its ability to train very deep networks with improved performance and easier optimization. Here is a high-level description of the ResNet-50 algorithm:

1. **Input Layer**: The input to ResNet-50 is an RGB image of size 224x224.

2. **Convolutional Layers**: ResNet-50 begins with a single convolutional layer with a large receptive field (7x7) and a stride of 2. This is followed by a batch normalization layer and a ReLU activation function. The stride of 2 reduces the spatial dimensions of the input by half.

3. **Max Pooling Layer**: After the initial convolutional layer, ResNet-50 applies a max pooling layer with a pool size of 3x3 and a stride of 2. This further reduces the spatial dimensions.

4. **Residual Blocks**: ResNet-50 consists of 16 residual blocks, each containing multiple convolutional layers. A residual block is composed of the following operations:

   (a) **Identity Block**: The identity block has three convolutional layers, each followed by a batch normalization layer and a ReLU activation function. The size of the filters is typically 1x1, 3x3, and 1x1, respectively. This block is used when the input and output dimensions are the same.

   (b) **Convolutional Block**: The convolutional block is similar to the identity block, but it includes a convolutional layer with a stride of 2 in the first 1x1 filter. This block is used when the input and output dimensions differ. Convolutional Block: The convolutional block is similar to the identity block, but it includes a convolutional layer with a stride of 2 in the first 1x1 filter. This block is used when the input and output dimensions differ.

   Each residual block takes the output from the previous block or layer as input and performs a series of operations, including identity or convolutional blocks, to compute the final output. The shortcut connection in each block allows the gradients to flow directly through the block, avoiding the problem of vanishing gradients.

5. **Global Average Pooling**: After the last residual block, ResNet-50 applies global average pooling to convert the spatial feature maps into a vector by taking the average of each feature map.

6. **Fully Connected Layer**: A fully connected layer with a softmax activation function is added at the end to produce the final class probabilities. The number of units in this layer corresponds to the number of classes in the dataset.

7. **Loss Function**: The standard loss function used with ResNet-50 for image classification tasks is the cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the true labels.

8. **Training**: ResNet-50 is trained using backpropagation and gradient descent optimization. During training, the model learns the optimal weights for the convolutional layers and the fully connected layer.

9. **Inference**: Once trained, ResNet-50 can be used to make predictions on new images by passing them through the network and obtaining the class probabilities from the softmax layer.

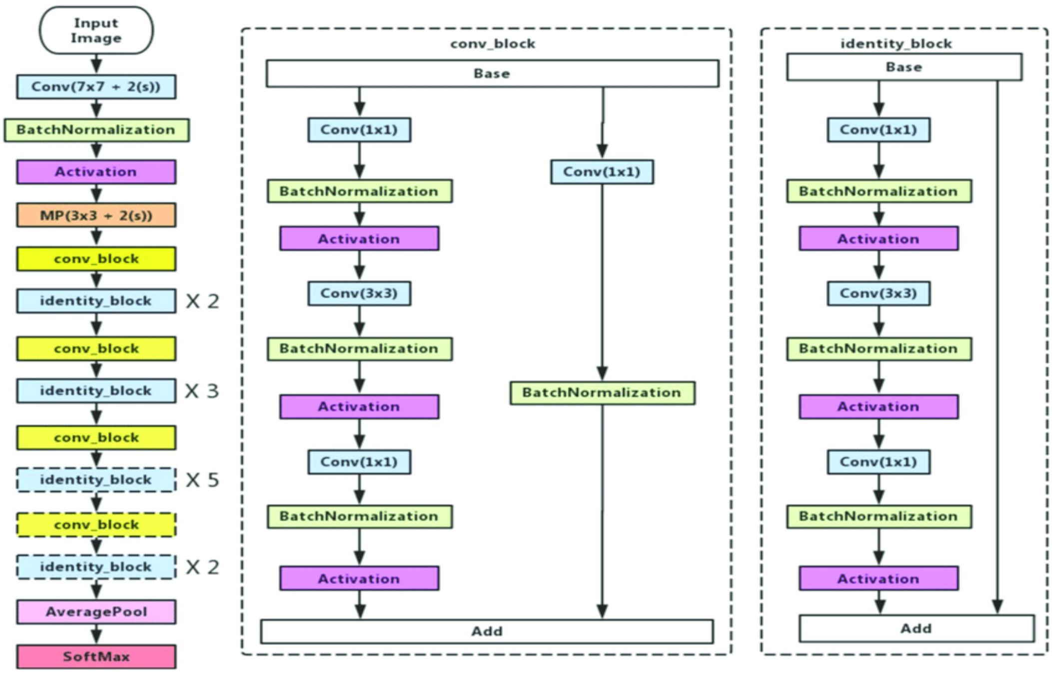

Figure 6.2: ResNet50 Architecture

CHAPTER 7

SOFTWARE TESTING

7.1 Type of Testing

- Unit Testing: This type of testing focuses on verifying the individual components or units of code in isolation. It aims to validate that each unit functions correctly and meets its expected behavior.
- Integration Testing: Integration testing is conducted to test the interaction and communication between different modules or components of a software system. It ensures that the integrated system functions as expected and that the modules work together properly.
- Functional Testing: Functional testing verifies that the software application meets the specified functional requirements. It tests the system against functional specifications and ensures that it performs its intended tasks correctly.
- Performance Testing: Performance testing evaluates the performance and scalability of the software application under different load conditions. It measures response times, throughput, resource usage, and identifies performance bottlenecks.
- Usability Testing: Usability testing assesses how user-friendly and intuitive the software application is. It involves evaluating the software's ease of use, navigation, and overall user experience.

7.2 Test cases & Test Results

- Test Case ID:1
- Test Case Description:The fundus image of an eye that carries no disease.
- Expected Results: Normal
- Test Data: Fundus image
- Actual Results: Normal
- Status: Pass

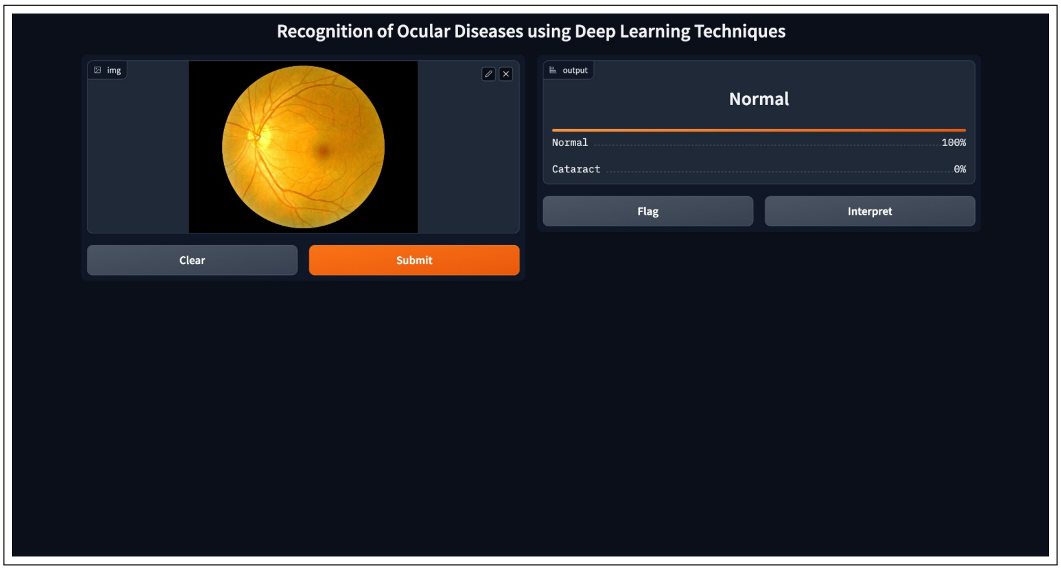

Figure 7.1: Test case

CHAPTER 8

RESULTS

8.1 Outcomes

1. The primary outcome of this project is the successful development of the ocular disease detection software solution. This outcome involves designing, coding, and testing the software to use VGG-16 model giving us an accuracy of 90.625%.

2. Bug-free and reliable software that caters to the needs of medical professionals. It meets the functional requirements to ease the process of eye image scanning and result delivery.

8.2 Screen Shots

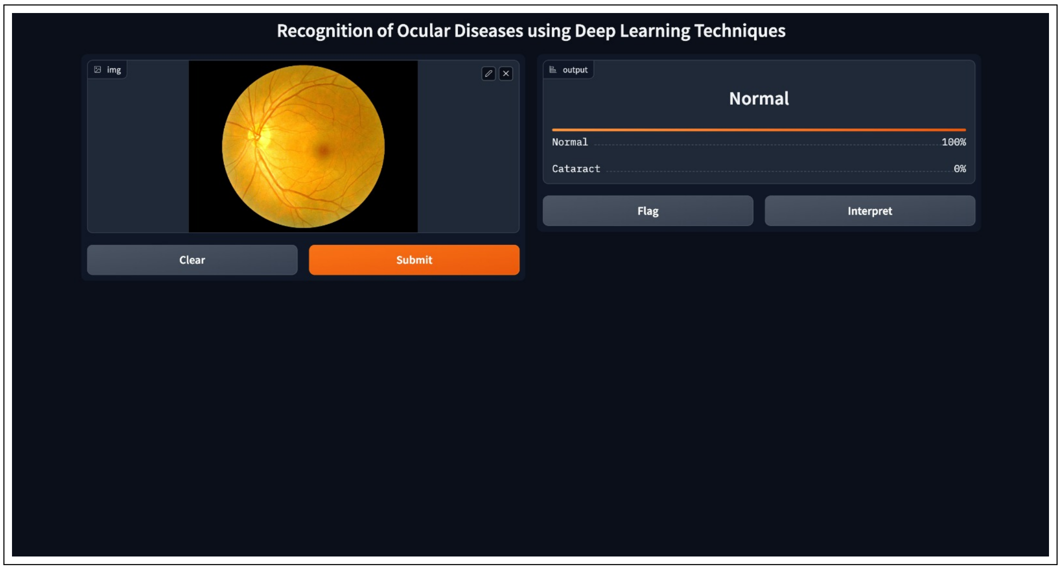

Figure 8.1: Test case

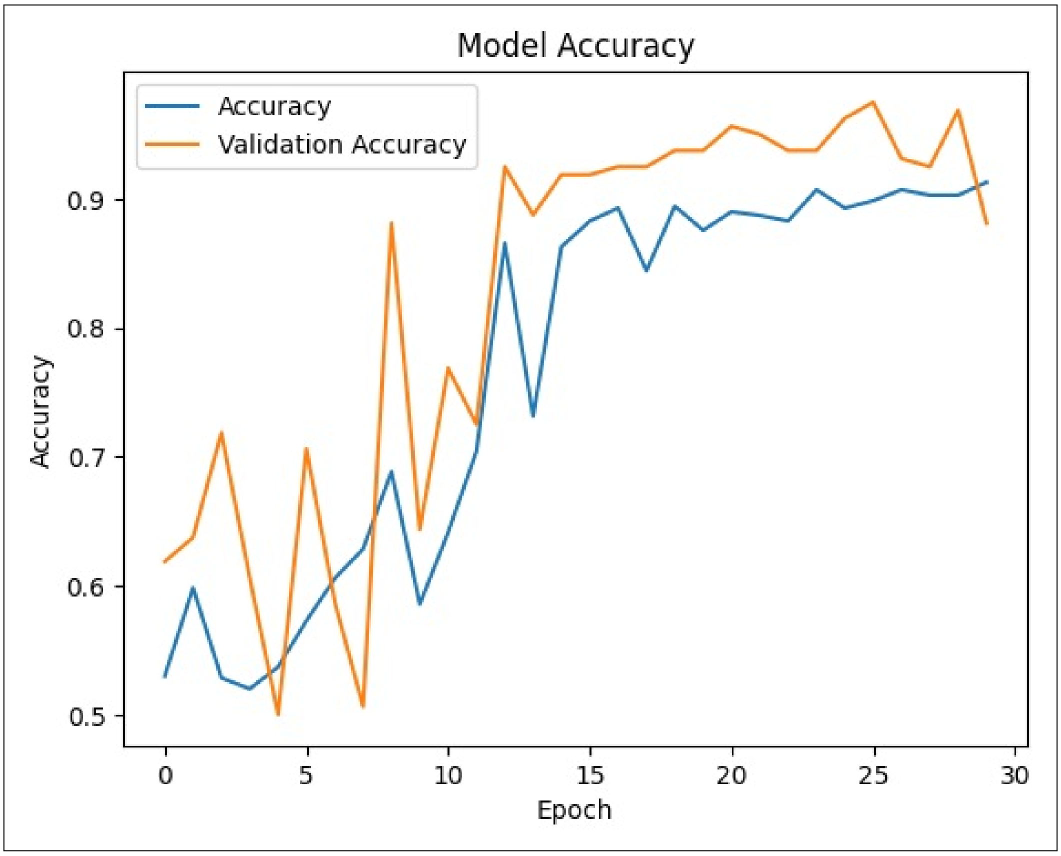

Figure 8.2: Model Accuracy

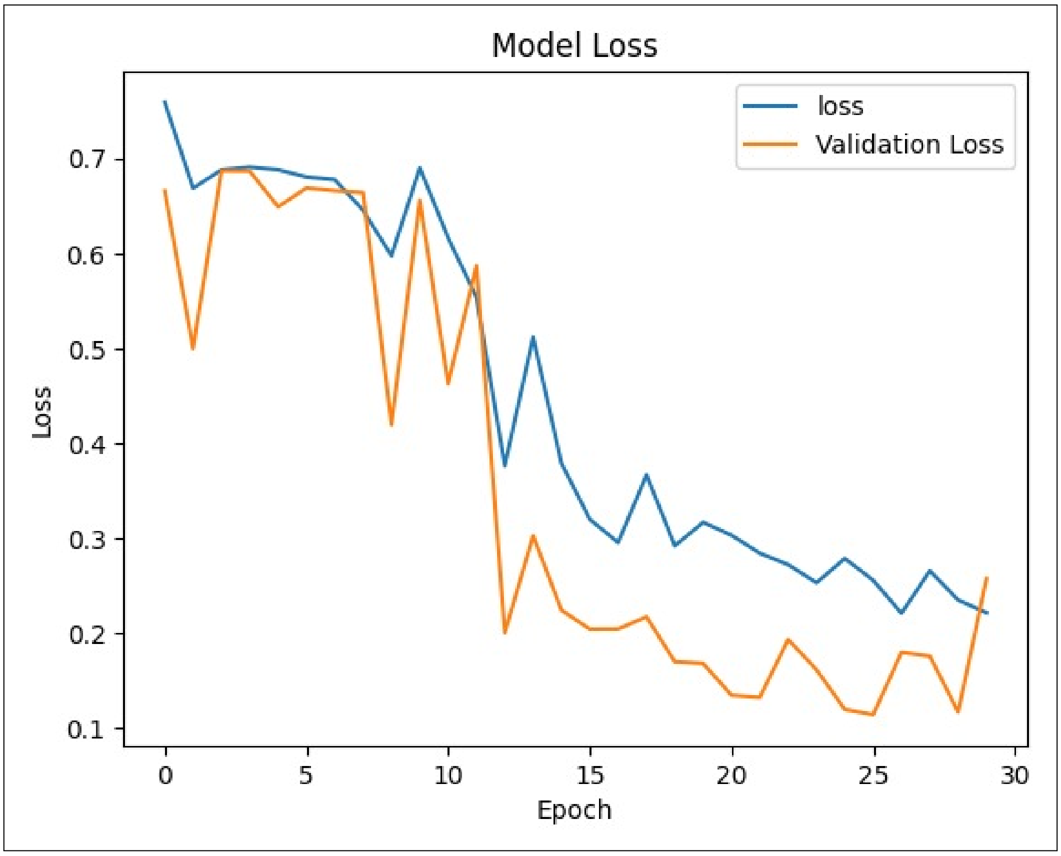

Figure 8.3: Model loss

CHAPTER 9

CONCLUSIONS

9.1 Conclusions

In conclusion, this project holds significant potential in advancing the field of eye care and improving patient outcomes. The development and implementation of robust ocular disease detection systems have the power to revolutionize the way eye diseases are diagnosed, monitored, and treated. By leveraging advancements in machine learning, image analysis algorithms, and data-driven approaches, these projects have the capability to detect and classify various ocular diseases accurately and at an early stage. This early detection can enable timely interventions, leading to better treatment outcomes and preservation of vision. We have implemented the VGG-16 module under CNN giving us an accuracy of 90.625 %.

9.2 Future Work

1. Dataset inclusivity : Improvement in dataset inclusivity by making it more generalized and spread across over geographical regions.

2. Enhanced Accuracy and Performance: Continued research and development can focus on improving the accuracy and performance of ocular disease detection algorithms and models. This can involve exploring advanced machine learning techniques, incorporating deep learning algorithms, and leveraging large datasets to train and fine-tune models for better detection and classification accuracy.

3. Integration with Emerging Technologies: Ocular disease detection projects can be integrated with emerging technologies to enhance their capabilities. For example, combining ocular disease detection with artificial intelligence (AI) assistants or virtual reality (VR) platforms can provide real-time guidance, interactive visualizations, and personalized recommendations for patients and healthcare professionals.

4. Mobile and Wearable Solutions: Mobile and wearable technologies offer a promising future scope for ocular disease detection. Developing smartphone applications or wearable devices with integrated sensors for capturing eye images or conducting quick eye screenings can provide convenient and accessible tools for early detection and monitoring of ocular diseases.

5. Predictive Analytics and Risk Assessment: Future projects can focus on incorporating predictive analytics and risk assessment models into

ocular disease detection systems. By leveraging historical data and patient-specific factors, these systems can identify individuals at higher risk of developing specific ocular conditions and provide personalized recommendations for preventive measures and early intervention.

6. Collaborative Platforms and Knowledge Sharing: Developing collaborative platforms and knowledge-sharing networks can facilitate the exchange of expertise, data, and insights among researchers, clinicians, and industry professionals. These platforms can accelerate innovation, foster collaboration, and enable collective learning to drive advancements in ocular disease detection and treatment.

9.3 Applications

1. Early Detection and Diagnosis: An eye disease detection system can help in the early detection and diagnosis of various eye conditions and diseases, such as diabetic retinopathy, glaucoma, macular degeneration, and cataracts. Early detection allows for timely treatment and management, improving the chances of preserving vision and preventing further damage.

2. Screening Programs: Eye disease detection projects can be integrated into screening programs, especially in areas with limited access to ophthalmologists or healthcare facilities. These projects can be used to identify individuals at risk of eye diseases and refer them for further examination and treatment.

3. Telemedicine and Remote Diagnosis: Eye disease detection systems can be incorporated into telemedicine platforms, enabling remote diagnosis and consultation. Patients can capture images or videos of their eyes using a smartphone or camera, which can be analyzed by the system to provide preliminary assessments and recommendations.

4. Public Health Campaigns: Eye disease detection projects can support public health campaigns aimed at raising awareness about common eye conditions and promoting regular eye check-ups. These projects can provide accessible and user-friendly tools for individuals to assess their eye health and encourage them to seek professional care when necessary.

5. Research and Data Analysis: Eye disease detection projects generate a wealth of data that can be used for research and analysis. Aggregated

and anonymized data from these projects can help identify patterns, trends, and risk factors associated with different eye diseases, contributing to advancements in understanding and treatment.

6. Assistive Technologies: Eye disease detection systems can be integrated into assistive technologies to assist visually impaired individuals. For example, the system can provide real-time feedback and guidance on the presence of obstacles or hazards, enhancing mobility and independence.

7. Clinical Decision Support: Eye disease detection projects can serve as a clinical decision support tool for healthcare professionals. By analyzing images or scans of the eye, the system can provide insights and recommendations to assist ophthalmologists and optometrists in making accurate diagnoses and treatment plans.

8. Monitoring and Progress Tracking: Eye disease detection projects can be used to monitor the progression of eye diseases over time. By regularly capturing and analyzing images, the system can track changes in the eye and help healthcare providers assess the effectiveness of treatments or interventions.

CHAPTER 10

APPENDIX

10.1 Appendix A

- Technical Feasibility: Assess the technical feasibility of solving the problem statement. The required technologies, tools, and expertise to develop the proposed solution are available in the form of collaborative notebooks and python libraries.
- Economic Feasibility:The costs involved in developing and maintaining the solution, including development resources, hardware, software licenses, infrastructure, and ongoing operational costs is minimal and affordable. Most cost can be attributed to resource optimization.
- Schedule Feasibility: The project can be completed within the desired timeframe owing to appropriate planning.

CHAPTER 11
REFERENCES

[1] Li, N., Li, T., Hu, C., Wang, K., Kang, H. (2021). A Benchmark of Ocular Disease Intelligent Recognition: One Shot for Multi-disease Detection. In: Wolf, F., Gao, W. (eds) Benchmarking, Measuring, and Optimizing. Bench 2020. Lecture Notes in Computer Science(), vol 12614. *Springer*, Cham. https://doi.org/10.1007/978-3-030-71058-3_11

[2] Md Shakib Khan, Nafisa Tafshir, Kazi Nabiul Alam, Abdur Rab Dhruba, Mohammad Monirujjaman Khan, Amani Abdulrahman Albraikan, Faris A. Almalki, Deep Learning for Ocular Disease Recognition: An Inner-Class Balance. *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 5007111, 12 pages, 2022.

[3] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2015). Rethinking the Inception Architecture for Computer Vision. *arXiv*. https://doi.org/10.48550/arXiv.1512.00567

[4] Gulshan V, Peng L, Coram M, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*. 2016;316(22):2402–2410. doi:10.1001/jama.2016.17216

[5] Sarki, Rubina Ahmed, Khandakar Wang, Hua Zhang, Yanchun. (2020). Automated detection of mild and multi-class diabetic eye diseases using deep learning. *Health Information Science and Systems*. 8. 10.1007/s13755-020-00125-5

[6] Elloumi, Y., Akil, M., Boudegga, H. (2019). Ocular Diseases Diagnosis in Fundus Images using a Deep Learning: Approaches, tools and Performance evaluation. *arXiv*. https://doi.org/10.48550/arXiv.1905.02544

[7] Islam, M. T., Imran, S. A., Arefeen, A., Hasan, M., Shahnaz, C. (2019). Source and Camera Independent Ophthalmic Disease Recognition from Fundus Image Using Neural Network. *2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)*. doi:10.1109/spicscon48833.2019.9065162

[8] X. Luo, J. Li, M. Chen, X. Yang and X. Li, ”Ophthalmic Disease Detection via Deep Learning With a Novel Mixture Loss Function,” in *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 9, pp. 3332-3339, Sept. 2021.

[9] WHO: World report on vision. World Health Organization (2019): https://www.who.int/publications-detail/world-report-on-vision

[10] Kaggle Ocular Disease Recognition dataset description:
https://www.kaggle.com/andrewmvd/ocular-disease-recognition-odir5k