DIABETIC RETINOPATHY DETECTION SYSTEM PROJECT DOCUMENTATION

1. Introduction

Project Title:

Diabetic Retinopathy Detection System Using Deep Learning

Team Members:

- **Team Leader:** Srinivas Surisetti – Responsible for overall project planning, CNN model development, integration, documentation, and deployment.
- **Team Member:** Assisted in backend integration using Python and Flask, API routing, and system testing.
- **Team Member:** Worked on frontend development using HTML, CSS, Bootstrap, and JavaScript.
- **Team Member:** Contributed to dataset preprocessing, model evaluation, and performance optimization.

Diabetic Retinopathy (DR) is a major complication of diabetes and one of the leading causes of preventable blindness worldwide. Manual diagnosis requires trained ophthalmologists to examine retinal fundus images, which is time-consuming and not scalable in rural or underdeveloped regions.

This project introduces an automated screening system using Deep Learning to classify retinal images as **Infected (DR Present)** or **Uninfected (Healthy)**. A Convolutional Neural Network (CNN) is trained on retinal fundus images and deployed through a Flask-based web application to provide instant predictions.

1.2 Purpose

The primary purpose of this project is to design and implement an intelligent medical image classification system capable of assisting in early screening of Diabetic Retinopathy.

Manual DR screening requires trained ophthalmologists and is:

- Time-consuming
- Expensive
- Limited in rural accessibility
- Prone to fatigue-related diagnostic errors

This project aims to:

- Provide automated screening support
- Reduce diagnosis time

- Improve accessibility in remote areas
- Demonstrate practical application of Deep Learning in healthcare

---

2. IDEATION PHASE

2.1 Problem Statement

Diabetic Retinopathy is one of the leading causes of preventable blindness worldwide. Early detection is critical, but access to ophthalmologists is limited, especially in rural and underdeveloped regions.

The current manual screening system:

- Requires expert evaluation
- Is not scalable
- Increases healthcare costs
- Delays diagnosis

There is a need for an automated, scalable, and reliable AI-based screening solution that can assist in early detection of DR using retinal fundus images.

2.2 Empathy Map Canvas

Users:

- Ophthalmologists
- Rural healthcare workers
- Diabetic patients

User Pain Points:

- Limited specialist availability
- Long waiting times
- High screening costs
- Risk of missed early-stage detection

User Needs:

- Fast diagnosis
- Affordable screening
- Easy-to-use system
- Reliable decision support

2.3 Brainstorming

During ideation, the following solutions were considered:

- Manual rule-based image analysis
- Traditional machine learning classifiers (SVM, Random Forest)
- Convolutional Neural Networks (CNN)
- Transfer Learning using pre-trained models (ResNet, VGG)

CNN was selected due to:

- Strong performance in image classification
- Ability to extract spatial features automatically
- Better scalability for medical image tasks

---

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens web application
2. Uploads retinal image
3. System preprocesses image
4. CNN model performs prediction
5. Result displayed with confidence score
6. User receives screening outcome

3.2 Solution Requirement

Functional Requirements:

- Upload retinal images (JPG, PNG, TIFF)
- Perform preprocessing (resize to 64×64, normalization)
- Load trained CNN model
- Display classification result
- Show confidence percentage

Non-Functional Requirements:

- Fast response time (2–5 seconds)
- User-friendly interface
- Reliable predictions
- Lightweight deployment

3.3 Data Flow Diagram

Fig. 2: Level 0 Context DFD of the Diabetic Retinopathy Detection System

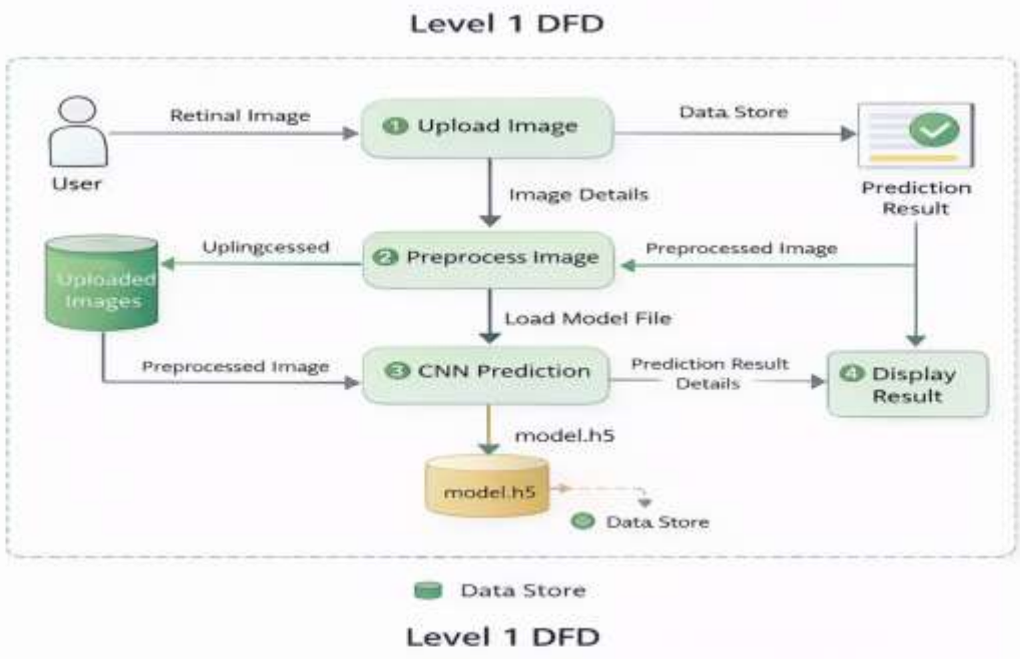

Fig. 3: Level 1 DFD of the Diabetic Retinopathy Detection System

Step 1: User uploads retinal image
Step 2: Flask saves image
Step 3: TIFF conversion (if required)
Step 4: Resize image to 64×64 pixels
Step 5: Normalize pixel values (0–1 range)
Step 6: CNN model inference
Step 7: Class label mapping
Step 8: Result displayed to user

3.4 Technology Stack

Machine Learning:

- Python 3.8
- TensorFlow
- Keras
- NumPy
- OpenCV

Web Development:

- Flask
- HTML
- CSS
- Bootstrap 5
- JavaScript

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

The shortage of trained ophthalmologists and increasing diabetic population require scalable screening tools.

This solution fits the problem because:

- AI replicates pattern recognition capability
- Provides quick screening results
- Reduces manual workload
- Supports rural healthcare outreach

4.2 Proposed Solution

The proposed solution is a CNN-based binary image classification system deployed as a web application.

The system:

- Accepts retinal image input
- Processes image through trained CNN
- Outputs classification (Infected/Uninfected)
- Displays confidence score

4.3 Solution Architecture

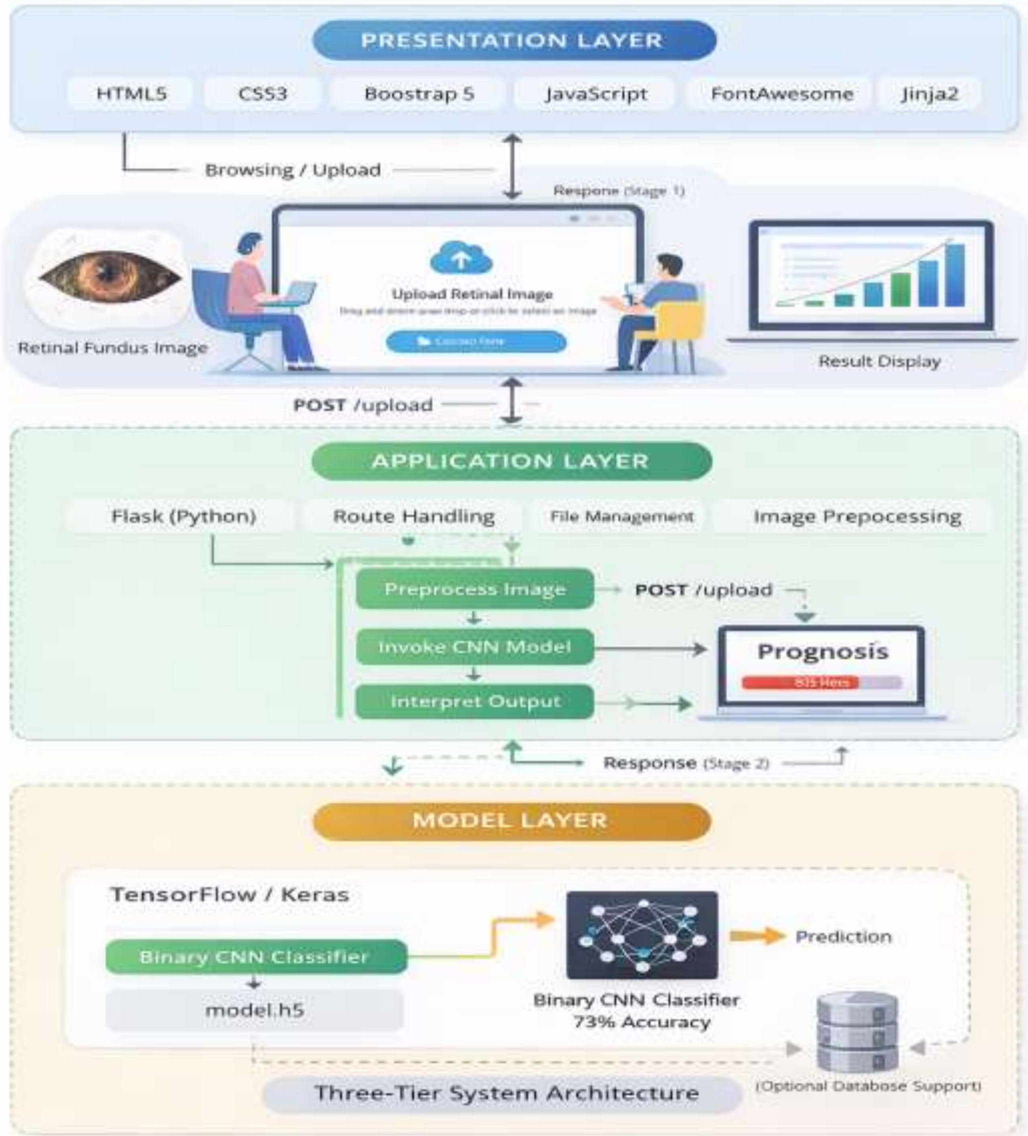

Fig. 1: Architecture Diagram of the " Diabetic Retinopathy Detection System

The system follows a **Three-Tier Architecture**:

1. Presentation Layer – HTML, CSS, Bootstrap frontend
2. Application Layer – Flask backend handling routing and preprocessing
3. Model Layer – TensorFlow/Keras CNN model for prediction

This modular design improves maintainability and scalability.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Phase 1: Dataset Collection & Preprocessing
Phase 2: CNN Model Design
Phase 3: Model Training & Evaluation
Phase 4: Web Application Development
Phase 5: Integration & Testing
Phase 6: Documentation & Deployment

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- Model Accuracy: **73%**
- Training Epochs: 10
- Batch Size: 32
- Inference Time: 2–5 seconds

Functional Testing:

- Upload validation
- TIFF format support
- Prediction output validation
- Error handling testing
- Server startup testing

The model showed consistent convergence with no major overfitting observed.

---

7. RESULTS

The system successfully classifies retinal images into infected or uninfected categories.

- Achieved 73% validation accuracy
- Correctly identifies most visible DR patterns
- Provides confidence-based prediction output

7.1 Output Screenshots

Figure 1: Home Page – Retinal Image Upload Interface

The above figure shows the main interface of the Diabetic Retinopathy Detection System. The user is provided with a drag-and-drop upload area to submit retinal fundus images for analysis. The interface supports multiple image formats including JPG, PNG, and TIFF. The "Choose File" and "Analyze Image" buttons enable easy interaction for non-technical users.

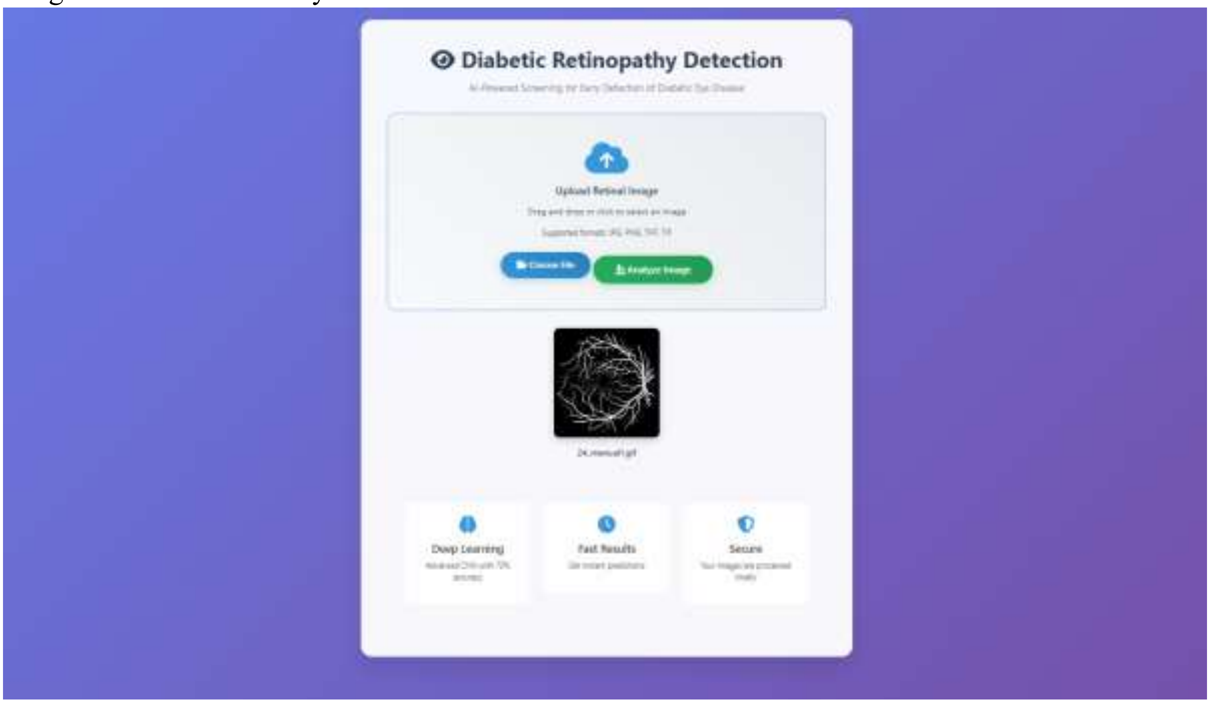

Figure 2: Image Selected Before Analysis

This figure displays the uploaded retinal image preview before analysis. Once the user selects a file, the system enables the "Analyze Image" button. This preview ensures that the correct image is selected before sending it to the backend for preprocessing and CNN-based classification.

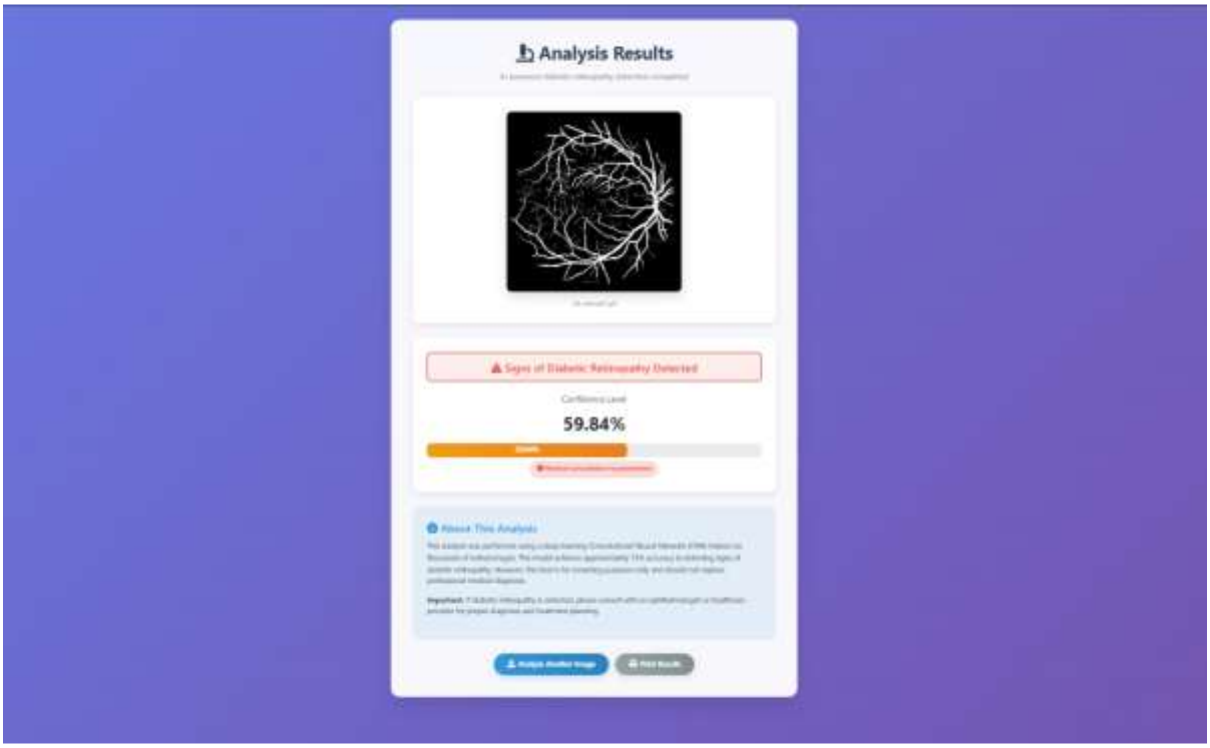

Figure 3: Prediction Result – Infected Case

The above figure shows the prediction results page generated by the CNN model. The system detected signs of Diabetic Retinopathy with a confidence level of 59.84%. The result is displayed with a highlighted warning message and a confidence bar visualization. A medical consultation recommendation is also provided as a precautionary advisory. The page includes options to analyze another image or print the result.

---

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Automated screening system
- Fast diagnosis
- Reduces manual workload
- Accessible and scalable
- Easy-to-use interface

Disadvantages:

- Only binary classification
- 73% accuracy not clinically sufficient
- Low image resolution (64×64)
- No cloud deployment
- No patient history storage

9. CONCLUSION

The Diabetic Retinopathy Detection System successfully demonstrates an end-to-end Deep Learning pipeline for automated medical image screening.

The system integrates CNN-based classification with a Flask web application to provide real-time predictions. Although the current accuracy is 73%, the project establishes a strong foundation for future improvements using transfer learning and higher-resolution models.

The project highlights the potential of AI in improving healthcare accessibility and reducing preventable blindness.

---

10. FUTURE SCOPE

- Implement Transfer Learning (ResNet50, EfficientNet)
- Multi-class DR severity grading
- Increase input resolution to 224×224
- Grad-CAM visualization for explainability
- Cloud deployment (AWS/GCP)
- REST API integration
- Database for patient history
- Mobile application development

---

11. APPENDIX

Source Code (if any):

- CNN Model Training Script (Model.py)
- Flask Application (app.py)
- HTML Templates & Static Files

Dataset Link:

APTOS 2019 Blindness Detection Dataset (Kaggle)
https://www.kaggle.com/c/aptos2019-blindness-detection

GitHub & Project Demo Link: https://github.com/SrinivasSurisetti/DiabeticRetinopathyDetection **&** https://drive.google.com/file/d/1CZDHOjNSUqGLX976KRLorHxjDdSZhE-5/view?usp=sharing