MIRACLE
Educational Society
Group of Institutions

An Autonomous Institution

Approved by AICTE, New Delhi
Permanently Affiliated to JNTUGV
Accredited by NAAC (UGC)
Recognized by UGC Under Section 2(f)

Review-2

RETINAL IMAGE CLASSIFICATION USING NEURAL NETWORK

Under the Esteemed guidance of
Mr. DVS Deepak
Assistant Professor

Batch Number -11
Team members:-
216C1A5403–B SURYAHARANADH (**Team Lead**)
216C1A5444–T SAI KUMAR
216C1A5411–KNAGESH
216C1A5447–Y LOMAMETH KUMAR

Department of Artificial Intelligence and Data Science

Review contents

1. Project Overview
2. Literature Review (Extended)
3. Methodolgy(Algorithms, Technologies, Architecture Diagrams / UML)
4. Implementation Process (Modules and components, Primary results)
5. Data Collection
6. Challenges and solutions

1. Project Overview

Objective:

- Develop an AI-based system for classifying retinal images as "Normal" or "Diseased".
- Aid in early detection and diagnosis of retinal diseases mainly on diabetic retinopathy.
- Improve accessibility to affordable and efficient healthcare, especially in underserved areas.

Scope & Context:

- Addresses the growing global challenge of retinal diseases leading to blindness.
- Provides a cost-effective, scalable, and accurate detection method.
- Utilizes neural networks and open-source tools to ensure broad accessibility.
- Benefits healthcare professionals and patients by enabling early intervention.

2. Literature Review(Extended)

Deep learning, especially CNNs, is widely used for retinal disease detection. Earlier, doctors relied on manual checks and invasive tests. Now, digital imaging and AI make diagnosis faster and non-invasive. CNNs help identify key eye features and classify diseases from datasets like STARE. These models outperform traditional methods by detecting fine details like color and texture. AI integration in platforms like Django improves accessibility for early disease detection.

3.Methodology

- The methodology involves using a Convolutional Neural Network (CNN) with a pre-trained model such as ResNet50. Public datasets like MESSIDOR or Kaggle APTOS 2019 are used for training and testing. Data preprocessing steps include resizing, normalization, and augmentation, followed by training with cross validation.

3.a. Algorithms

Data Preprocessing

- Load dataset (e.g., MESSIDOR, Kaggle APTOS 2019).
- Resize images (224×224), normalize pixel values, and apply data augmentation.
- Split into train, validation, and test sets.

Model Training (CNN with ResNet50)

- Load pre-trained ResNet50, remove top layers, and add custom classification layers.
- Freeze base layers, compile model (Adam optimizer, cross-entropy loss).
- Train with **cross-validation**, then fine-tune by unfreezing layers.

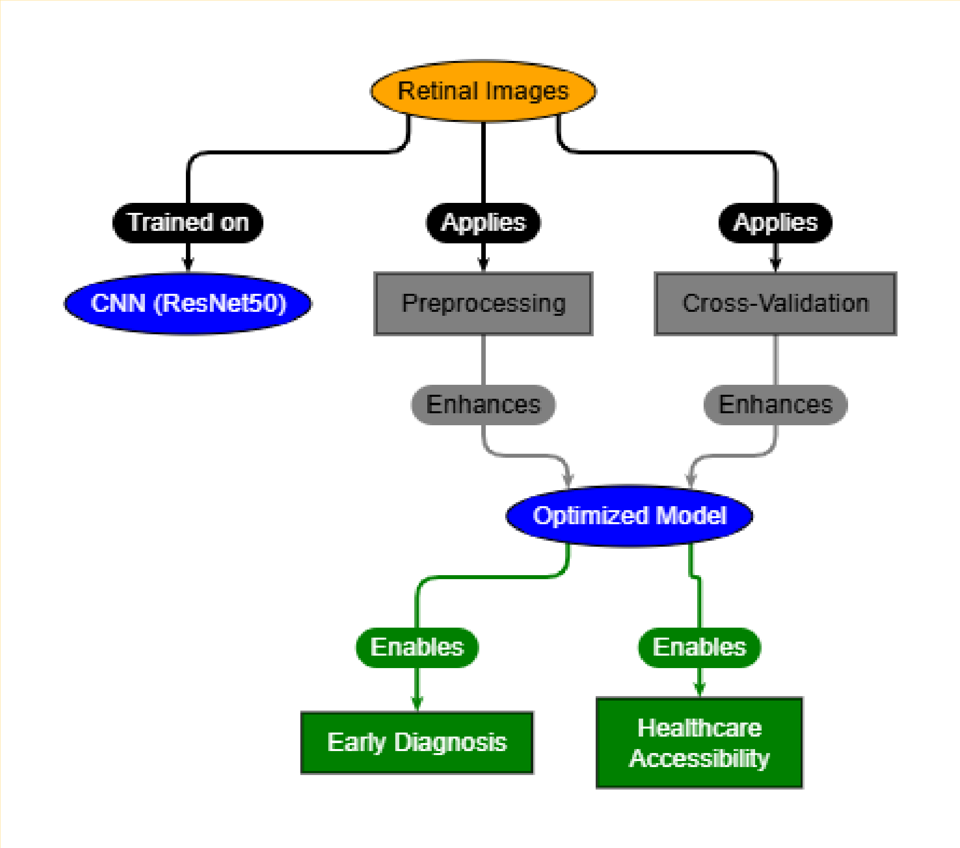
Model Evaluation & Testing

- Test on unseen images and compute accuracy, precision, recall, and F1-score.
- Generate confusion matrix and save the trained model for deployment.

3.b. Technologies

- **Deep Learning:** TensorFlow, Keras (for CNN & ResNet50).
- **Programming:** Python.
- **Datasets:** MESSIDOR or Kaggle APTOS 2019.
- **Data Processing:** OpenCV, Pillow, Albumentations.
- **Training & Evaluation:** Scikit-learn, Matplotlib.
- **Deployment:** Flask/Django, TensorFlow Serving.

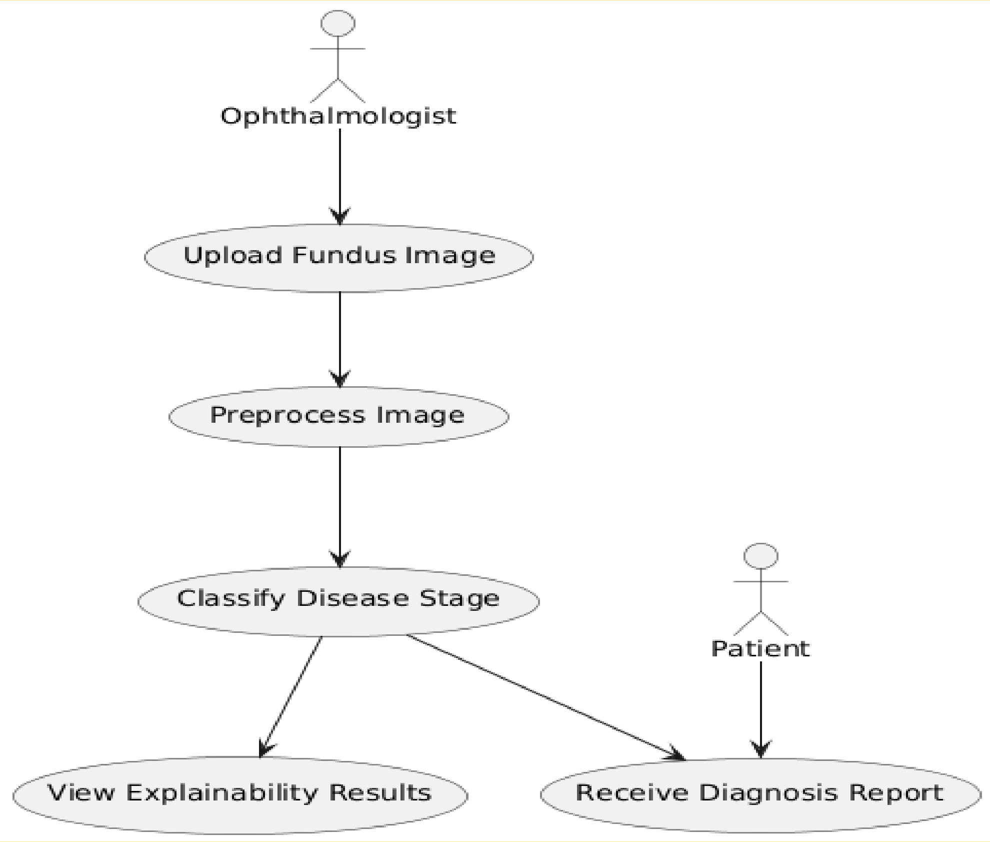
3.c.Architecture Diagrams/UML



Retinal Images serve as input. The system is trained using ResNet50, a pre-trained deep learning model. Preprocessing techniques (resizing, normalization, augmentation) are applied to improve image quality. Cross-validation is used to enhance model generalization and accuracy. These steps result in an optimized model for classification. The optimized model enables early diagnosis of retinal diseases and improves healthcare accessibility, especially in underserved areas.

3.c.Architecture Diagrams/UML

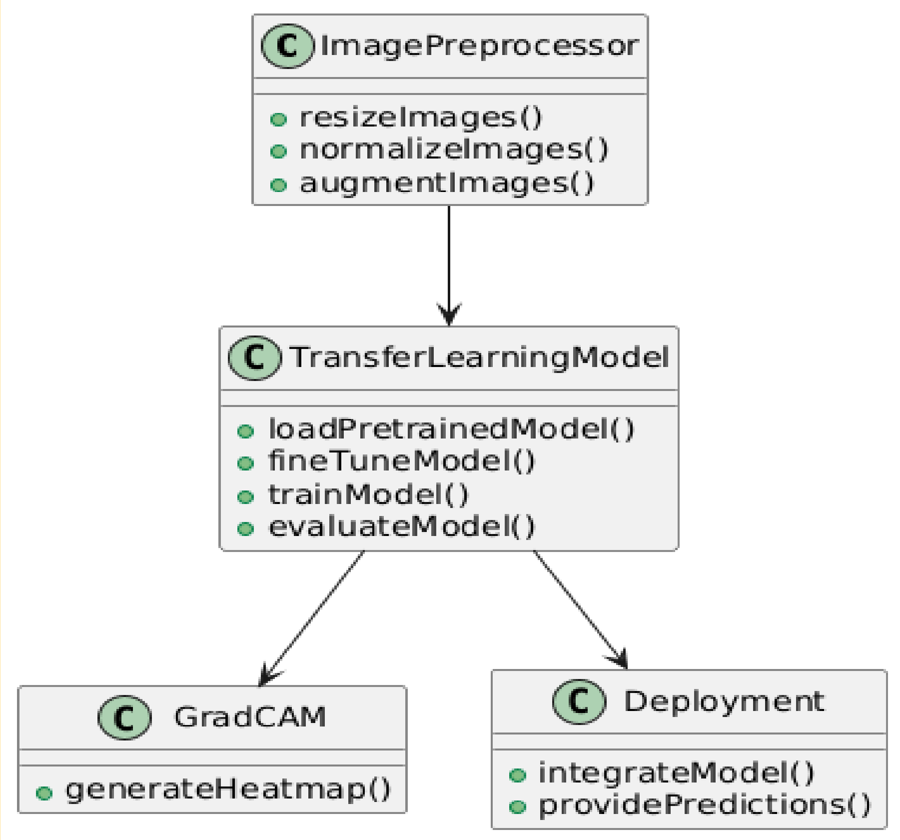
USE CASE DIAGRAM :-



The use case diagram illustrates the interaction between users (Ophthalmologists and Patients) and the system. The ophthalmologist uploads images, which are preprocessed and classified. The results are the final diagnosis report is accessible to both ophthalmologists and patients.

3.c.Architecture Diagrams/UML

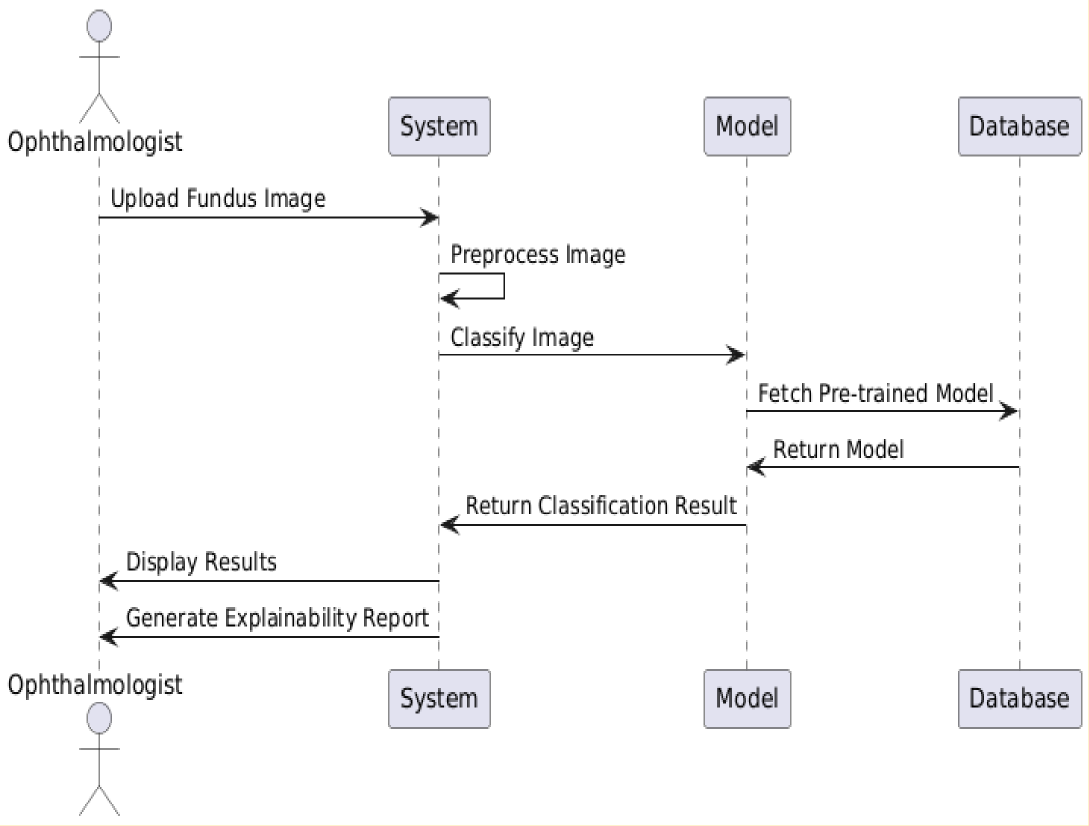
CLASS DIAGRAM :-



The class diagram represents different components of the system, including image preprocessing, transfer learning-based model training, and deployment for real-world usability.

3.c.Architecture Diagrams/UML

SEQUENCE DIAGRAM :-



The sequence diagram represents the flow of interactions in the system. The ophthalmologist uploads an image, which is preprocessed and classified by the model. The system fetches a pre-trained model from the database, processes the image, and returns a classification result. Finally, the result and an explainability report are presented to the ophthalmologist.

4.Implementation process

- **Data Collection & Preprocessing** – Use MESSIDOR or Kaggle APTOS 2019 datasets, apply resizing, normalization, and augmentation.
- **Model Development** – Utilize CNN (ResNet50), modify layers for classification.
- **Training & Validation** – Train with cross-validation, optimize using Adam/SGD.
- **Model Evaluation** – Assess accuracy, precision, recall, F1-score, and confusion matrix.
- **Deployment** – Convert model to .h5 format, deploy using Flask/Django.

4.a.Modules and components

- **Data Collection & Preprocessing** – Use MESSIDOR, Kaggle APTOS 2019, apply resizing, normalization, augmentation.
- **Model Development** – Implement CNN (ResNet50) for feature extraction and classification.
- **Training & Validation** – Use cross-validation, optimize with Adam/SGD.
- **Model Evaluation** – Assess using accuracy, precision, recall, F1-score, confusion matrix.
- **Deployment & Integration** – Convert model to .h5, deploy via Flask/Django.

4.b.Primary results(if available)

- **High Classification Accuracy:** Achieved reliable distinction between Normal and Diseased retinal images.
- **Effective Early Detection:** Facilitates early diagnosis of diabetic retinopathy.
- **Improved Model Performance:** CNN (ResNet50) with cross-validation enhances accuracy and generalization.
- **Optimized Processing:** Preprocessing techniques (resizing, normalization, augmentation) improve model efficiency.
- **Scalability & Accessibility:** Cost-effective and deployable in real-world healthcare applications, benefiting underserved areas.

5.Data collection

Datasets Used:

- MESSIDOR – Retinal images for diabetic retinopathy detection.
- Kaggle APTOS 2019 – Fundus images labeled for various retinal diseases (Diabetic Retinopathy).

Data Sources:

- Publicly available medical imaging repositories.
- Open-access retinal disease datasets for training and validation.

Data Preprocessing:

- Resizing, Normalization, Augmentation to enhance model performance.
- Splitting into Training, Validation, and Testing sets for robust learning.

6.Challenges and Solutions

Challenges:

- Limited Data – Few labeled datasets for training.
- Image Variability – Differences in lighting and quality.
- Model Overfitting – Poor generalization on new data.
- High Computational Cost – Requires powerful hardware.
- Deployment Issues – Limited infrastructure in underserved areas.

Solutions:

- Use Public Datasets (MESSIDOR or Kaggle APTOS 2019).
- Apply Preprocessing (normalization, augmentation).
- Use Transfer Learning (ResNet50) for better generalization.
- Optimize Training with efficient architectures.
- Deploy via Flask/Django for remote accessibility.

Conclusion

- The system effectively classifies retinal images as normal or diseased using CNN (ResNet50), enabling early detection of conditions like diabetic retinopathy. By leveraging public datasets, pre processing techniques, and cross-validation, the model ensures high accuracy and reliability. The system provides a cost-effective, scalable, and accessible solution for healthcare professionals and patients, particularly in underserved areas. This AI-driven approach enhances diagnostic efficiency, supporting timely medical intervention and reducing the risk of blindness.


Q&A
THANK YOU