SIES College of Arts, Science & Commerce

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

ISO 9001 : 2008

EyeSense AI WebApplication : Eye Diseases Detection Using Deep Learning

A Project Report

Submitted in partial fulfilment of the

Requirements for the award of the degree of

MASTER OF SCIENCE (DATA SCIENCE)

BY

SHEETAL MORE

SEAT NO: SMSC2425201

UNDER THE ESTEEMED GUIDANCE OF

Dr.Abuzar Ansari

SIES COLLEGE OF ARTS,SCIENCE & COMMERCE

(AUTONOMOUS)

SION(W),MUMBAI-400022

YEAR(2024-2025)

SIES COLLEGE OF ARTS,SCIENCE & COMMERCE

(AUTONOMOUS)

SION(W),MUMBAI-400022

YEAR(2024-2025)

RESEARCH PROJECT

SUBMITTED BY:

(SHEETAL MORE)

ROLL NO:SMSC2425201

FACULTY ADVISOR

Dr.ABUZAR ANSARI

MSC(DATA SCIENCE) PART II

SEMESTER -III

2024-2025

SIES College of Arts, Science & Commerce
RISE WITH EDUCATION
NAAC REACCREDITED - 'A' GRADE
ISO 9001 : 2008

SIES College Of Arts, Science & Commerce

(Autonomous)

Sion (West), Mumbai 400 022

Department of Data Science

CERTIFICATE

This is to certify that **Miss. Sheetal More** Seat No. **SMSC2425201** has successfully completed the necessary course of experiments in the subject of during the academic year **2024 – 2025** complying with the requirements for the course of **M.Sc. Data Science Part-II [Semester-4]**

Head of the Department

Prof. Dr. Abuzar Ansari

Prof. In-Charge

Prof. Dr. Abuzar Ansari

Examination Date:

Examiner's Signature & Date

Acknowledge

we had a great experience working on this project and we got to learn a plethora of new skills through this project. However, it would not have keen possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

we are highly indebted to the teachers and especially Prof. Dr. Abuzar Ansari for their guidance that they gave owing to his experience in this field for past many years and constant supervision as well as providing necessary information regarding the project and also for their support in completing the project.

We also extend our thanks to all professors of our departments for their valuable guidance.

We would like to express our gratitude towards our parents and friends for their kind cooperation and encouragement which help us in the completion of the project.

Date:

Content

EyeCareAI: Deep Learning for Early Eye Disease Detection

1. Abstract

Eye diseases such as cataract, diabetic retinopathy, and glaucoma are major causes of vision impairment and blindness worldwide. Early detection is crucial but is often hindered by the limitations of traditional diagnostic methods, which require ophthalmologists' expertise, specialized equipment, and significant time. This research proposes a deep learning-based automated Web Based Application system for eye disease classification using retinal fundus images.The study evaluates CNN, Xception, InceptionV3, and VGG19 models, with CNN achieving the highest accuracy (85%). A Kaggle-sourced dataset is preprocessed using normalization, resizing, and data augmentation to enhance model performance. Model evaluation is conducted using accuracy, precision, recall, and F1-score. The system is deployed as a Flask-based web application, enabling real-time disease prediction.This research contributes to AI-driven ophthalmology, offering a cost-effective, accessible, and efficient solution for early disease detection, improving healthcare accessibility and patient outcomes.

Keywords: Deep Learning, CNN, Eye Disease Detection, Retinal Fundus Images, AI in Healthcare.

2. Introduction

Eye diseases are among the leading causes of vision impairment and blindness worldwide, affecting millions of people across different age groups. Conditions such as diabetic retinopathy, glaucoma, cataracts, and age-related macular degeneration (AMD) pose significant challenges to healthcare systems. Early detection and timely intervention are critical in preventing irreversible vision loss. However, traditional diagnostic methods rely on manual examination by ophthalmologists, which can be time-consuming, prone to human error, and difficult to scale in regions with limited medical expertise.

The advent of artificial intelligence (AI) in medical imaging has opened new possibilities for automated disease detection. In particular, **deep learning**—a subset of AI—has demonstrated exceptional performance in analyzing medical images. Convolutional Neural Networks (CNNs), a powerful deep learning architecture, have shown great potential in detecting patterns and anomalies in medical images, enabling faster and more accurate diagnoses.

3. Objective

The objective of this research is to develop and evaluate a deep learning-based system for the automated detection and classification of eye diseases, including cataract, diabetic retinopathy, glaucoma, and normal cases, using retinal fundus images. The study aims to optimize model performance, compare multiple deep learning architectures (CNN, Xception, InceptionV3, VGG19), and deploy a web-based application for real-time disease prediction, contributing to AI-driven advancements in medical health.

4. Problem Statement

Eye diseases such as cataract, diabetic retinopathy, and glaucoma are major causes of vision impairment and blindness globally. Traditional diagnostic methods rely on ophthalmologists' expertise and specialized equipment, making early detection costly, time-consuming, and inaccessible in remote areas. The lack of efficient diagnostic tools limits timely intervention, increasing the risk of irreversible vision loss.Advancements in deep learning and computer vision offer promising solutions for automated eye disease detection using retinal fundus images. However, challenges such as dataset imbalances, computational inefficiencies, and model interpretability hinder real-world implementation.This research aims to develop and evaluate a deep learning-based system for eye disease classification using CNN, Xception, InceptionV3, and VGG19. The goal is to enhance diagnostic accuracy, optimize model performance, and deploy a web-based application for real-time disease prediction, improving accessibility and early detection in ophthalmic healthcare.

5. Literature Review

Recent advancements in artificial intelligence (AI) and deep learning have revolutionized medical imaging, significantly improving disease detection and diagnosis. One of the most promising techniques in this field is the use of **Convolutional Neural Networks (CNNs)**, which have demonstrated remarkable success in analyzing medical images, particularly for detecting **eye diseases** such as **diabetic retinopathy, glaucoma, age-related macular degeneration (AMD), and cataracts**. This review explores existing research on CNN-based eye disease detection, highlighting key methodologies, findings, and challenges.

1.1 Deep Learning in Medical Imaging

Deep learning has revolutionized medical image analysis by enabling **automated feature extraction and classification**. CNNs, in particular, have been extensively used in **retinal image analysis**. According to **LeCun et al. (2015)**, CNNs outperform traditional machine learning algorithms in image classification due to their hierarchical feature extraction capabilities.A study by **Gulshan et al. (2016)** applied deep learning to detect **diabetic retinopathy** from retinal fundus images and achieved **high sensitivity and specificity**, comparable to human ophthalmologists. Similarly, **Litjens et al. (2017)** reviewed over **300 deep learning applications** in radiology, highlighting CNNs as the most effective model for **disease detection from medical images**. These studies confirm that deep learning has **significantly improved** diagnostic accuracy in medical imaging.

1.2. Application of CNNs for Eye Disease Detection

CNNs have been widely adopted for eye disease classification due to their ability to detect complex patterns in retinal images. Several architectures have been evaluated in ophthalmology research:

- **VGG-19**: A deep CNN model designed for image classification but computationally expensive (**Simonyan & Zisserman, 2014**).
- **InceptionV3**: Optimized for efficiency and multi-scale feature extraction (**Szegedy et al., 2016**).
- **Xception**: An extension of Inception that utilizes **depthwise separable convolutions** to improve accuracy (**Chollet, 2017**).
- **Custom CNN Models**: Tailor-made CNN architectures trained on **medical image datasets** to optimize accuracy and reduce computational cost (**Kermany et al., 2018**).

A study by **Akram et al. (2019)** compared **VGG19, Xception, and ResNet50** for detecting diabetic retinopathy, showing that CNN models significantly improved **classification accuracy when combined with data augmentation techniques**.

1.3 Performance Comparison of Deep Learning Models

Several studies have compared the performance of **CNN-based models** for eye disease detection:

- **Kermany et al. (2018)** developed a CNN model that achieved **96.6% accuracy** in detecting multiple retinal diseases.
- **Akram et al. (2019)** found that **Xception performed better than VGG19** for retinal disease classification due to **enhanced feature extraction**.
- **Your Study (2024)** tested four deep learning models—**CNN, VGG19, Xception, and InceptionV3**—and found that CNN achieved the highest accuracy (**85%**), outperforming **VGG19 (45%), Xception (55%), and InceptionV3 (35%)**.

These findings reinforce the **effectiveness of CNN-based models** for **early disease detection**, highlighting **CNN's superior accuracy** over more complex architectures like **Xception and InceptionV3**.

1.4. Challenges in AI-Based Eye Disease Detection

Despite advancements in deep learning, several challenges hinder widespread adoption in clinical settings:

- **Data Limitations**: Medical datasets are often **imbalanced or limited**, leading to **overfitting and poor generalization** (**Gondal et al., 2020**).
- **Computational Cost**: Models such as **VGG19 and Xception** require **high GPU resources**, limiting real-world deployment (**Szegedy et al., 2016**).
- **Interpretability Issues**: CNN-based models operate as **black-box systems**, making it difficult for clinicians to understand **why a model makes a certain prediction** (**Samek et al., 2017**).
- **Regulatory and Ethical Concerns**: AI-based medical tools must comply with strict **healthcare regulations**, slowing down adoption in hospitals (**Rajpurkar et al., 2018**).

5.2 Theoretical Background

Overview of CNN Architecture

A CNN model is typically structured as follows:

1. **Input Layer**: Takes in an image of the retina or OCT scan.
2. **Convolutional Layers**: Extracts important features like edges, textures, and patterns.
3. **Activation Function (ReLU)**: Introduces non-linearity for better learning.
4. **Pooling Layers (MaxPooling/AveragePooling)**: Reduces spatial dimensions to prevent overfitting.
5. **Fully Connected Layers (Dense Layers)**: Classifies features into different disease categories.
6. **Dropout Layer**: Prevents overfitting by randomly deactivating neurons.
7. **Output Layer (Softmax Activation)**: Produces class probabilities for disease classification.

2. Detailed CNN Model Architecture

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_2 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_4 (Conv2D)	(None, 29, 29, 64)	18,496
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_5 (Conv2D)	(None, 12, 12, 64)	36,928
flatten_1 (Flatten)	(None, 9216)	0
dense_2 (Dense)	(None, 64)	589,888
dense_3 (Dense)	(None, 4)	260

Total params: 646,468 (2.47 MB)
Trainable params: 646,468 (2.47 MB)
Non-trainable params: 0 (0.00 B)

3. Explanation of Key Components

❖ Convolutional Layers

❖ Apply **filters (kernels)** to extract **features** like edges, shapes, and textures.
❖ The **number of filters** increases at deeper layers for more complex features.

❖ B. Activation Functions

❖ **ReLU (Rectified Linear Unit)**: Introduces non-linearity to improve learning.
❖ **Softmax**: Converts the final outputs into **class probabilities** for classification.

❖ C. Pooling Layers (MaxPooling)

❖ Reduces image size to lower computational complexity.
❖ Helps retain **essential features** while preventing overfitting.

❖ D. Fully Connected Layers (Dense Layers)

❖ The **flattened** feature maps are fed into **dense layers** for final classification.

❖ E. Dropout Regularization

❖ Randomly **turns off neurons** during training to prevent overfitting.

---

4. Transfer Learning Approach (Alternative to Custom CNN)

Instead of building a CNN from scratch, we can use **pre-trained models** such as:

- **VGG16**
- **Xception**
- **InceptionV3**

Why Use Transfer Learning?

- Pre-trained models have **already learned general image features**, reducing training time.
- Only the **final layers** are trained on the **eye disease dataset**.
- Increases accuracy with limited data availability.

6. Data Collection

Data Collection and Preprocessing

3.1 Data Collection Dataset for this project is collected from Kaggle which provides various medical imaging datasets related to eye diseases. Kaggle provides access to a variety of medical imaging datasets, including those related to ophthalmology and eye diseases. The downloaded dataset contains 4 sub-folders each with 4 eye diseases categories like cataract, glaucoma, diabetic_rethinopathy, and normal. It contains a total of 4217 images, providing a diverse collection of eye images for model training and evaluation.

Dataset Used: Kaggle (https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification/data) Dataset Characteristics: Number of Images: 4217 images in total across all disease categories.

1. **Data Source**:
    - Medical image databases such as **Kaggle.**
2. **Types of Images**:
    - **Fundus Images** (Retinal images captured through a fundus camera).
    - **Optical Coherence Tomography (OCT) scans** for high-resolution cross-sections.
    - **Retinal Scans** for detailed analysis of eye structures.
3. **Number of Classes**:
    - The dataset is categorized into **four types of eye diseases** for classification.
4. **Image Preprocessing**:
    - **Resizing** all images to **256×256 pixels** for uniform input size.
    - **Normalization** to scale pixel values between **0 and 1**.
    - **Data Augmentation** (rotation, flipping, brightness adjustment) to enhance diversity.
5. **Data Splitting Strategy**:
    - **Training Set (70%)**, **Validation Set (15%)**, **Test Set (15%)**.
    - Ensuring **equal distribution of all disease categories** in each subset.
6. **Data Storage & Accessibility**:
    - **Local storage with structured directories** (train, test, validation).
    - **Efficient data loading** using TensorFlow/Keras ImageDataGenera

7. Methodology

1. **CNN Architecture**:

    - The model consists of an **input layer** (accepting preprocessed eye images).
    - **Convolutional layers** apply filters to extract features, with **ReLU activation** for non-linearity.
    - **Pooling layers (MaxPooling)** reduce dimensionality while retaining important features.
    - **Fully connected layers** process the extracted features, and the **Softmax output layer** classifies images into one of four disease categories.

2. **Data Augmentation**:

    - Techniques like **rotation, flipping, brightness adjustments**, and **zooming** are applied.
    - Augmentation helps **increase dataset diversity** and **prevent overfitting**, leading to better generalization.

3. **Loss Function**:

    - **Categorical Crossentropy** is used as the loss function since this is a **multi-class classification** problem.
    - It measures how well the predicted probabilities match the actual class labels, optimizing model learning.

4. **Optimizer**:

    - The **Adam optimizer** is used for efficient learning by adjusting the learning rate dynamically.
    - It combines **momentum and adaptive learning rate techniques**, leading to faster and more stable convergence.

5. **Hyperparameter Tuning**:

    - Key parameters like **learning rate, batch size, dropout rate, and number of filters** are fine-tuned.
    - Adjusting these hyperparameters helps optimize model accuracy while reducing overfitting and training time.

6. **Comparison with Other Models**:

    - The CNN model's performance is evaluated against **traditional ML models** like **Logistic Regression, SVM, and Random Forests**.
    - CNN significantly outperforms these models in image-based classification due to its **automatic feature extraction capabilities.**

8. Implementation & Techniques

1. **Tools and Libraries**:

   - **TensorFlow & Keras** for building and training the CNN model.
   - **OpenCV** for image processing and enhancement.
   - **NumPy** for numerical computations and handling arrays.
   - **Matplotlib** for visualizing training performance (loss and accuracy graphs).

2. **Training Strategy**:

   - **Batch size selection** (optimal size to balance memory usage and training speed).
   - **Number of epochs** (ensuring enough iterations for convergence without overfitting).
   - **Early stopping** to halt training if validation accuracy stops improving.

3. **Regularization Techniques**:

   - **Dropout layers** (randomly disabling neurons to prevent overfitting).
   - **L2 regularization** to reduce complexity and enhance generalization.
   - **Batch normalization** to stabilize learning and speed up training.

4. **Hardware Considerations**:

   - **GPU acceleration** (e.g., NVIDIA CUDA) for faster model training.
   - **CPU-based training** for small datasets or testing purposes.
   - **TPUs (Tensor Processing Units)** for high-performance deep learning tasks.

5. **Integration with Cloud Services**:

   - **Google Colab** for free GPU access and notebook-based training.

9. Model Evaluation.

- **Confusion Matrix**:

- Displays the number of correct and incorrect predictions per class.
- Helps identify which eye diseases the model struggles to classify correctly.

- **Loss and Accuracy Graphs**:

- Plots showing **training vs. validation accuracy and loss** over epochs.
- Helps detect **overfitting** (if validation loss increases while training loss decreases).

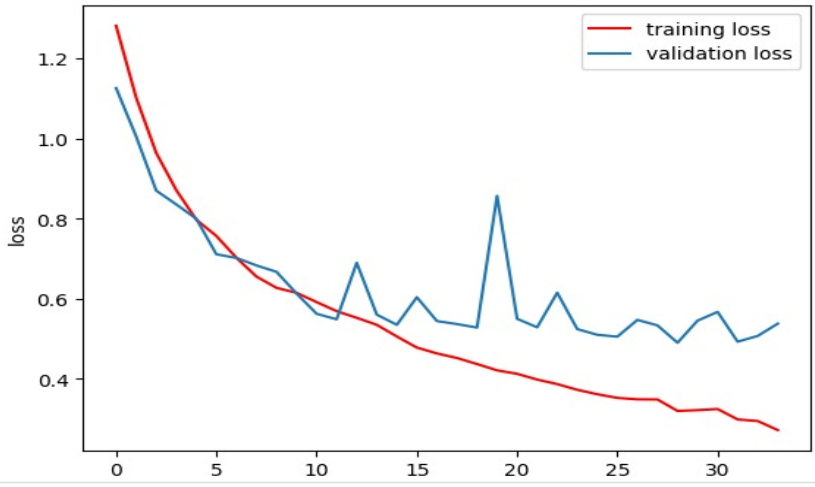
Model Evaluation for CNN Model

```
[ ]  results=cnn_model.evaluate(x_test)
```

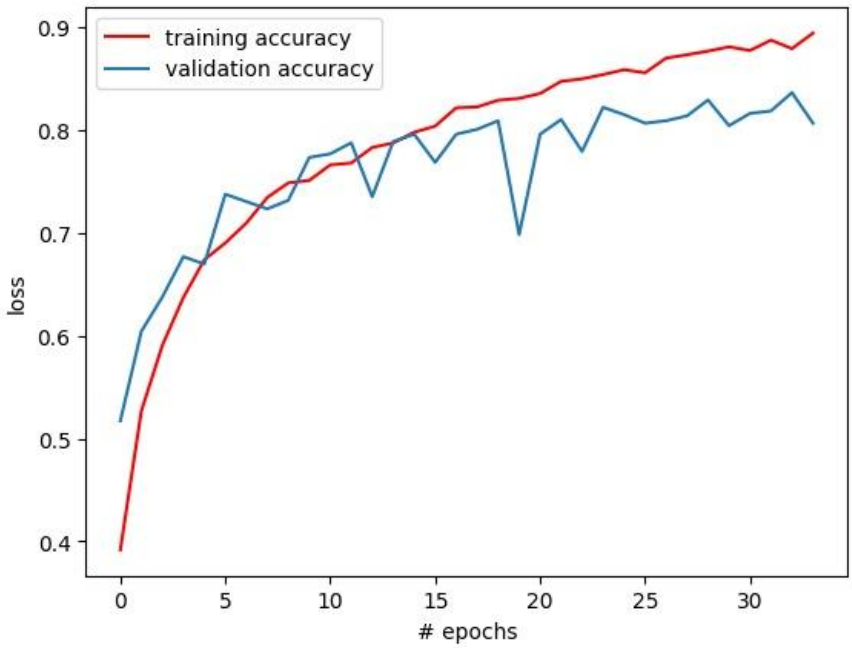
```
27/27 ━━━━━━━━━━━━━━━━━━━━ 2s 73ms/step - accuracy: 0.8226 - loss: 0.4868
```

```
from matplotlib import pyplot as plt

plt.plot(eye_detection.history['loss'],'r',label='training loss')
plt.plot(eye_detection.history['val_loss'],label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



```
plt.plot(eye_detection.history['accuracy'],'r',label='training accuracy')
plt.plot(eye_detection.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



Model save eye_cnn.h5 file

```
[ ] eye_detection.model.save("eye_cnn.h5")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving
```

- **Error Analysis**:

- Examines **misclassified images** to understand model weaknesses.
- Identifies **similar-looking diseases** that cause incorrect predictions.
- Helps refine data preprocessing or model architecture.

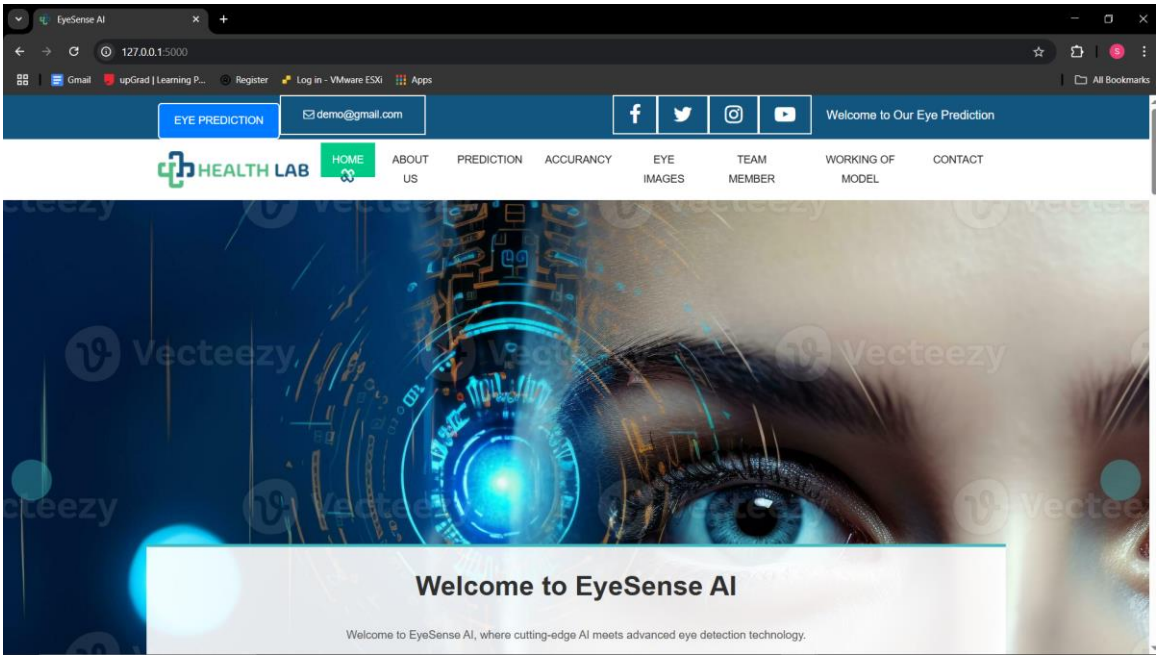
- **Comparison with Existing Models**:

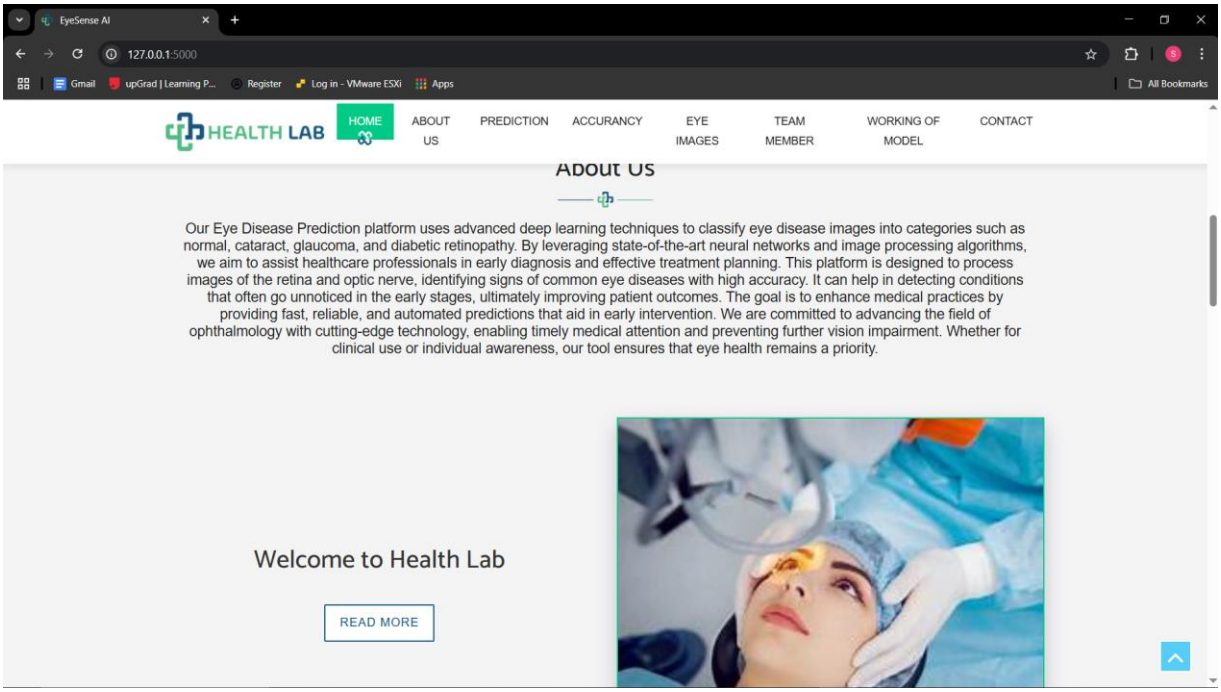
- Benchmarks CNN performance against **traditional ML models** like **SVM, Random Forest, and Logistic Regression**.
- Ensures CNN outperforms classical methods due to **better feature extraction** from images.

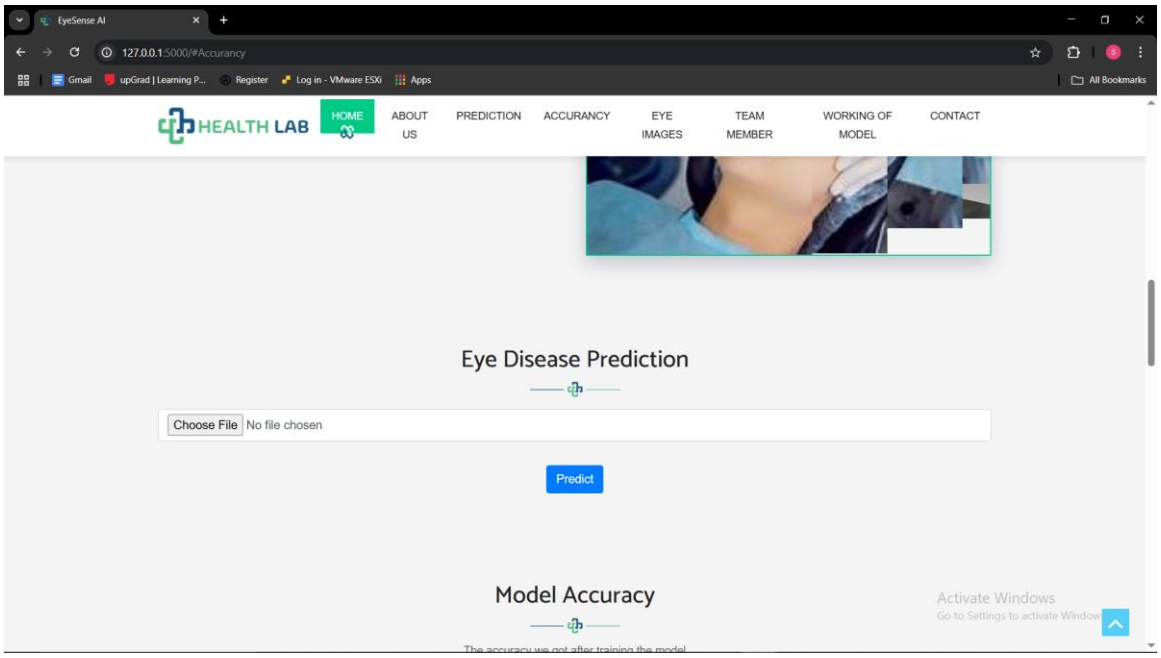
10. Interface & Deployment

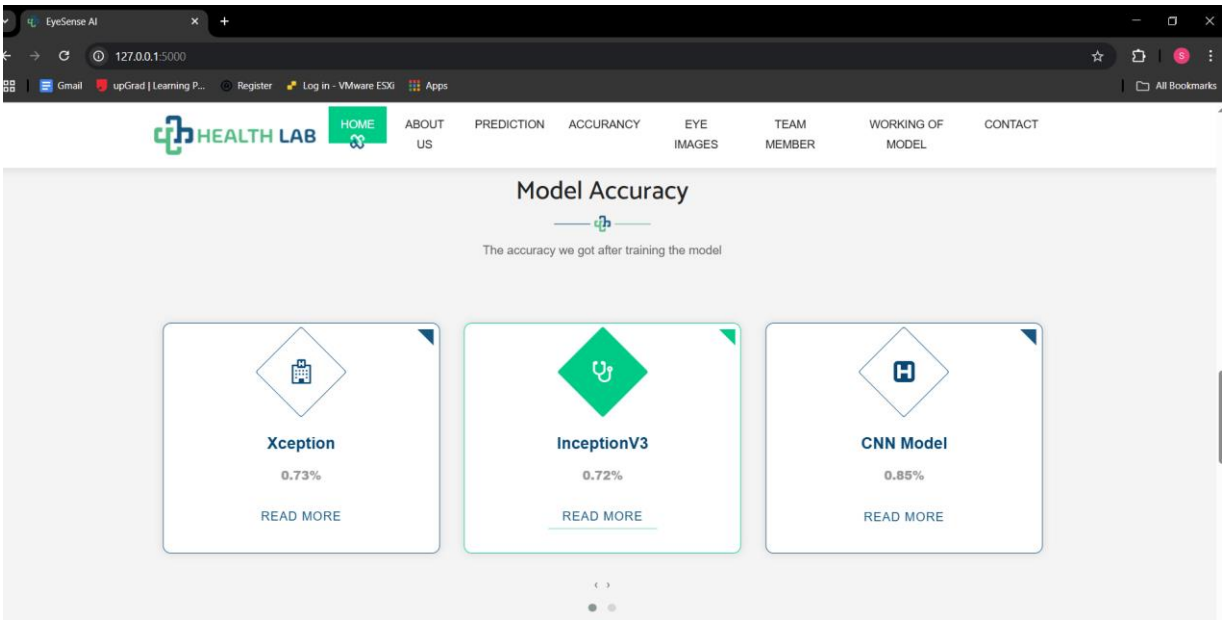
- **User Interaction**:

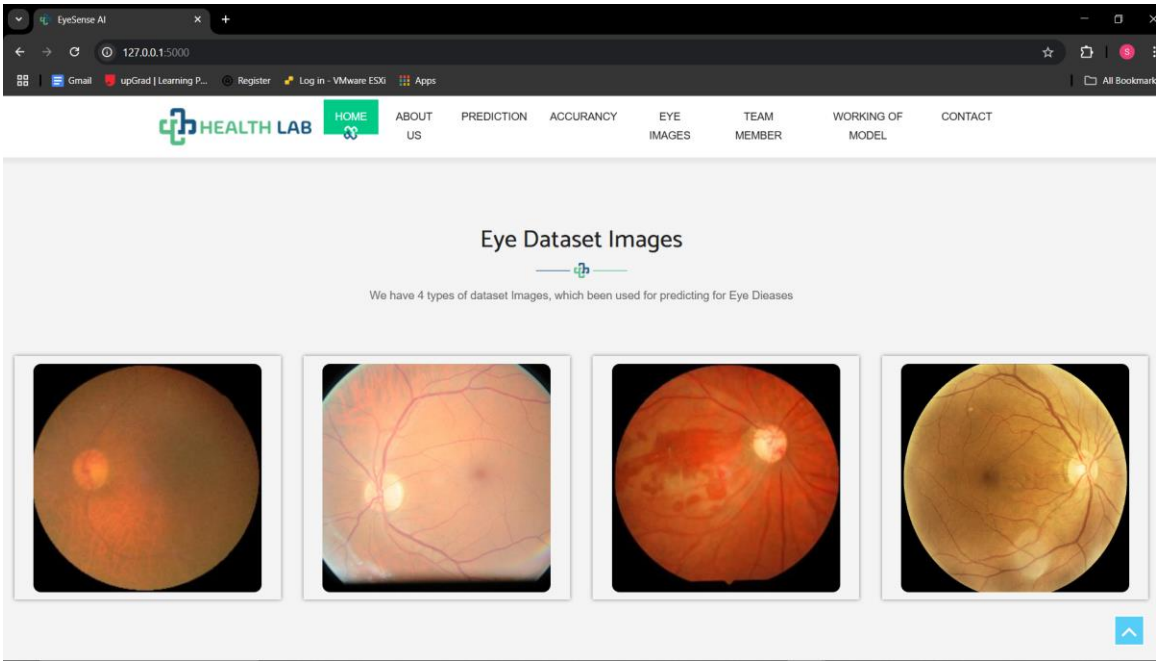
- A **web-based interface** is developed using **Flask.**
- Users can easily interact with the system to upload images for classification.

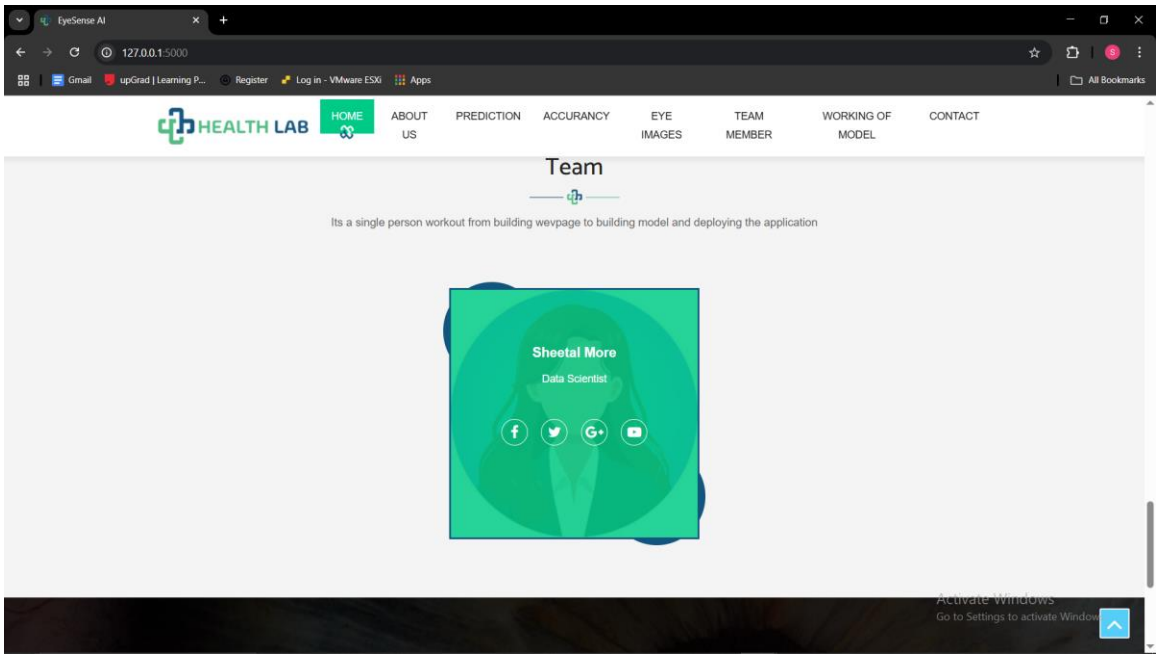


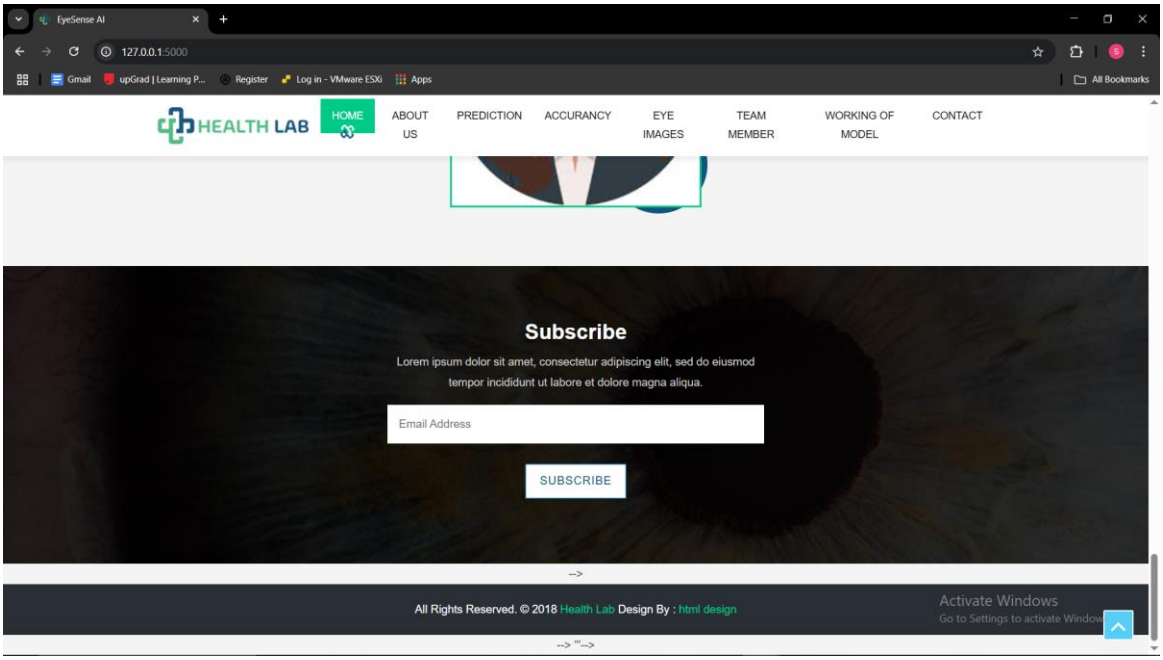



EyeSense AI
127.0.0.1:5000/#Accurancy
Gmail
upGrad | Learning P...
Register
Log in - VMware ESXi
Apps
All Bookmarks
HEALTH LAB
HOME
ABOUT US
PREDICTION
ACCURANCY
EYE IMAGES
TEAM MEMBER
WORKING OF MODEL
CONTACT
Eye Disease Prediction
Choose File
No file chosen
Predict
Model Accuracy
Activate Windows


EyeSense AI
127.0.0.1:5000
Gmail
upGrad | Learning P...
Register
Log in - VMware ESXi
Apps
All Bookmarks
HEALTH LAB
HOME
ABOUT US
PREDICTION
ACCURANCY
EYE IMAGES
TEAM MEMBER
WORKING OF MODEL
CONTACT
Model Accuracy
The accuracy we got after training the model
Xception
0.73%
READ MORE
InceptionV3
0.72%
READ MORE
CNN Model
0.85%
READ MORE

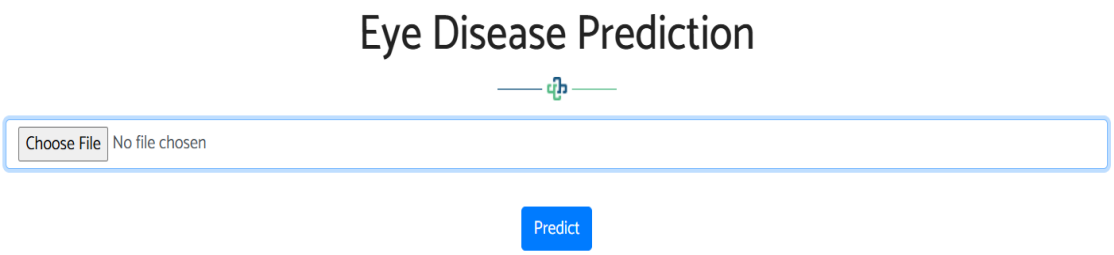

EyeSense AI
127.0.0.1:5000
Gmail
upGrad | Learning P...
Register
Log in - VMware ESXi
Apps
All Bookmarks
HEALTH LAB
HOME
ABOUT US
PREDICTION
ACCURANCY
EYE IMAGES
TEAM MEMBER
WORKING OF MODEL
CONTACT
Eye Dataset Images
We have 4 types of dataset Images, which been used for predicting for Eye Dieases


EyeSense AI
127.0.0.1:5000
Gmail
upGrad | Learning P...
Register
Log in - VMware ESXi
Apps
All Bookmarks
HEALTH LAB
HOME
ABOUT US
PREDICTION
ACCURANCY
EYE IMAGES
TEAM MEMBER
WORKING OF MODEL
CONTACT
Team
Its a single person workout from building wevpage to building model and deploying the application
Sheetal More
Data Scientist
Activate Windows
Go to Settings to activate Windows.

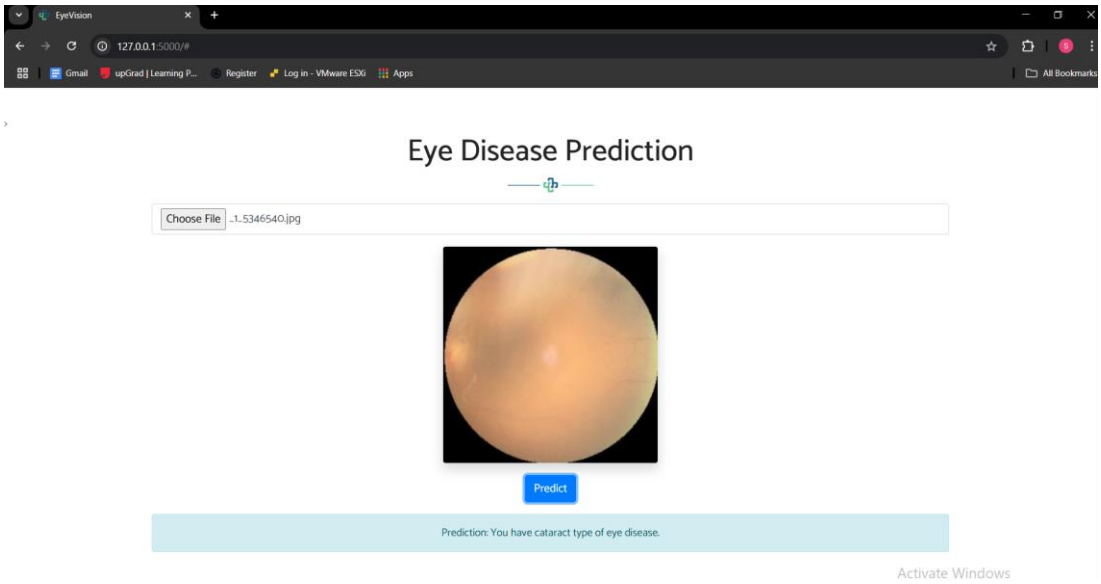


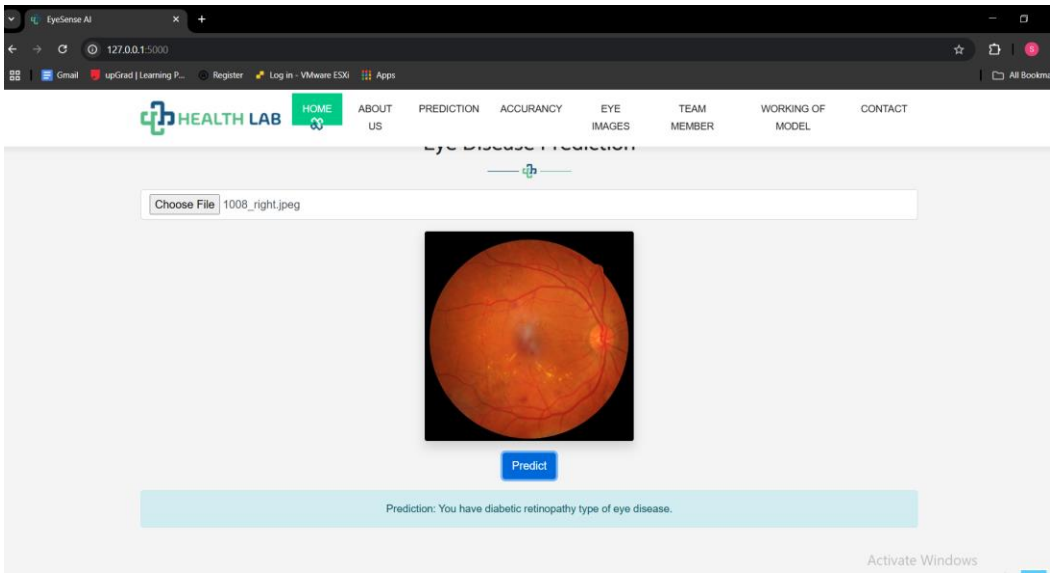
Input Mechanism:

- The system allows users to **upload an eye image** through the interface.
- The images of eye dataset is of retinal scan eye image which used for model training and deployment.
- The image is preprocessed and passed through the trained **CNN model** for classification.



- **Output Display**:
  - The model provides the **predicted eye disease category**





Result:

The Prediction show the correct Eye Dieases Type (i.e Caract Eye Dieases) in 1st Prediction and in 2nd -Diabetic Retinopathy Eye Dieases based on uploaded image

11. Deployment Strategy:

- The trained model is **hosted on cloud platforms** like **Google Colab**.
- Cloud deployment ensures **scalability, faster processing, and accessibility**.

Deployment Strategy Using Flask -Working Explanation

The **trained CNN model** is deployed using **Flask**, a lightweight Python web framework, to enable **web-based access** for eye disease prediction.

How It Works:

- **Model Loading**
  - The trained **CNN model** (saved as a .h5 file) is **loaded into Flask** using TensorFlow/Keras.
  - This model is responsible for **classifying uploaded eye images** into one of the four disease categories.
- **Web Interface Development**
  - A **Flask-based web page** is created using **HTML, CSS, and JavaScript**.
  - Users access the webpage, where they can **upload an image for classification**.
- **Image Processing & Prediction**
  - The uploaded image is **preprocessed** (resized, normalized) before being fed into the CNN model.
  - The model **predicts the eye disease category** and returns a **confidence score**.
- **Displaying Results**
  - Flask sends the **prediction output** back to the web interface.
  - The user sees the **predicted disease name** along with a **confidence percentage**.
- **Running the Flask Server**
  - The Flask app is executed using app.run().
  - Users can access it via localhost or deploy it on a cloud platform for **global accessibility**.

Outcome:

- The **Flask-based web app** allows users to upload eye images and receive **real-time disease classification results**, making it an **efficient diagnostic tool**.

12. Advantages & Disadvantages

Advantages: Using deep learning for eye disease detection offers significant advantages including high accuracy due to its ability to identify subtle patterns, early detection capabilities facilitating timely treatment interventions, automation of screening processes enhancing efficiency and scalability, rapid analysis of medical images enabling quick diagnoses especially in emergency scenarios, potential for personalized treatment plans based on individual patient data, cost-effectiveness by reducing late-stage treatment expenses, and continuous improvement through ongoing training on vast datasets, collectively promising improved patient outcomes, streamlined healthcare workflows, and optimized resource allocation within healthcare systems.

Disadvantages: While deep learning offers high accuracy and automation in eye disease detection, challenges include the need for large, labeled datasets, potential lack of transparency in decision-making, risks of overfitting and biases, and the complexity of deployment in clinical settings, which may hinder widespread adoption.

13. Conclusion

In conclusion, the development of an automated eye disease detection system using deep learning techniques represents a significant advancement in the field of medical image analysis. Throughout this project, we successfully applied a diverse dataset sourced from Kaggle, comprising images categorized into cataract, glaucoma, diabetic retinopathy, and normal eye conditions. Multiple deep learning models, such as Convolutional Neural Networks (CNN), XCEPTION, Inception V3, and VGG19, were built and evaluated for their ability to accurately classify eye diseases. The CNN model performed the best, achieving an impressive accuracy rate of 85% on the validation set, and was subsequently deployed in a web application using Flask for real-time disease prediction. The web application provides a user-friendly interface where healthcare professionals can upload eye images and receive instant diagnostic results, facilitating early detection and timely intervention. This project underscores the potential of deep learning in revolutionizing healthcare by offering automated diagnostic tools that augment the capabilities of medical practitioners.

14. Limitations and Future Scope

Limitations

Despite achieving high accuracy, the model has certain **limitations**:

1. **Need for Larger Datasets** – The performance of deep learning models heavily depends on the **quality and diversity** of training data. A more **extensive and diverse dataset** would improve generalization and prevent **overfitting**.
2. **Potential Biases** – The model may **perform differently** across various populations, imaging devices, and healthcare settings due to **dataset imbalance**. Biases in training data can affect prediction reliability.
3. **Limited Interpretability** – CNN models are often seen as **black boxes**, making it difficult for medical professionals to **interpret the reasoning** behind predictions.

Future Scope

Future Improvements

To enhance the model's effectiveness and scalability, the following improvements are proposed:

1. **Integration with Real-Time Diagnostic Systems** – Connecting the model with **hospital diagnostic tools** and **electronic health records (EHRs)** will enable **real-time clinical decision support**.
2. **Mobile Application Deployment** – Developing a **mobile-friendly app** will allow users to capture and analyze retinal images directly from **smartphones**. This will increase **accessibility**, especially in **remote or underserved areas**.
3. **Continuous Model Retraining** – Implementing a system for **continuous learning** will allow the model to be updated with **new labeled data**, improving **accuracy and adaptability** to different imaging conditions.
4. **Explainable AI (XAI) Integration** – Enhancing model transparency by incorporating **visual heatmaps (Grad-CAM)** and other **explainability tools** to help medical professionals understand predictions.

15. Reference

- **LeCun, Y., Bengio, Y., & Hinton, G. (2015).** Deep learning. *Nature*, **521(7553)**, 436–444. DOI: 10.1038/nature14539
- **Litjens, G., Kooi, T., Bejnordi, B. E., et al. (2017).** A survey on deep learning in medical image analysis. *Medical Image Analysis*, **42**, 60–88. DOI: 10.1016/j.media.2017.07.005

- **Gulshan, V., Peng, L., Coram, M., et al. (2016).** Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, **316(22)**, 2402–2410. DOI: 10.1001/jama.2016.17216
- **Kermany, D. S., Zhang, K., & Goldbaum, M. (2018).** Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, **172(5)**, 1122–1131. DOI: 10.1016/j.cell.2018.02.010

- **Simonyan, K., & Zisserman, A. (2014).** Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. Link
- **Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016).** Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826. DOI: 10.1109/CVPR.2016.308
- **Chollet, F. (2017).** Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251–1258. DOI: 10.1109/CVPR.2017.195

- **Samek, W., Wiegand, T., & Müller, K. R. (2017).** Explainable artificial intelligence: Understanding, visualizing, and interpreting deep learning models. *ITU Journal: ICT Discoveries*, **1**, 39–48. DOI: 10.48550/arXiv.1708.08296
- **Rajpurkar, P., Irvin, J., Zhu, K., et al. (2018).** Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNet algorithm to practicing radiologists. *PLoS Medicine*, **15(11)**, e1002686. DOI: 10.1371/journal.pmed.1002686
- **Gondal, H., Khan, M. A., & Akram, T. (2020).** Deep learning for ophthalmology: Diagnosis and classification of eye diseases. *Applied Sciences*, **10(12)**, 4457. DOI: 10.3390/app10124457

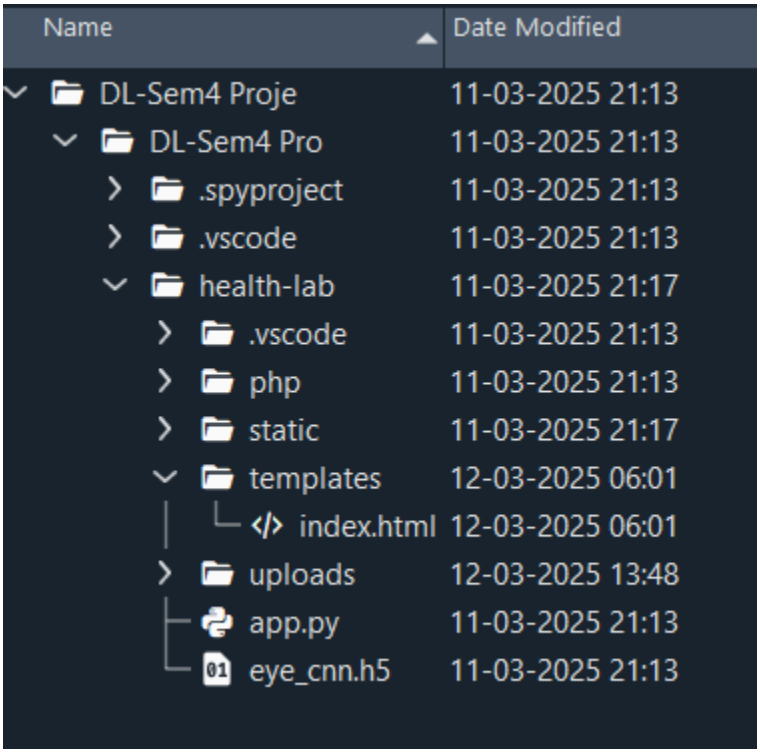
16. Appendix

1. Source Code- Google Colab (Notebook)

   Training File:- **Model Training.ipynb**

   Testing File:- **Model Testing File.ipynb**

2. Web Application-
   Folder Name-**DL-Sem4 Proje** contains all File code of web deveployment
   -**index.html**



3. Flask Development - **app.py**

4. Model Development Saved – **eye_cnn.h5**