PROJECT NAME: RETINANET

GROUP
GROUP 16

TEAM MEMBERS
PALANISELVAM PRIYANKA (Team Leader) - A0307250R
DHARSHINI CHELLAPPA CHETTY RAJAN – A0307202X
SWETA PATTNAIK - A0296537M

Table of Contents

Introduction

Access to quality medical care remains a critical challenge in many developing countries, particularly in specialized areas like ophthalmology. Inadequate resources and a shortage of trained professionals often result in significant delays in diagnosis and treatment, leading to the progression of preventable or treatable eye conditions. This not only exacerbates the burden of vision-related disabilities but also impacts the overall health and productivity of affected individuals.

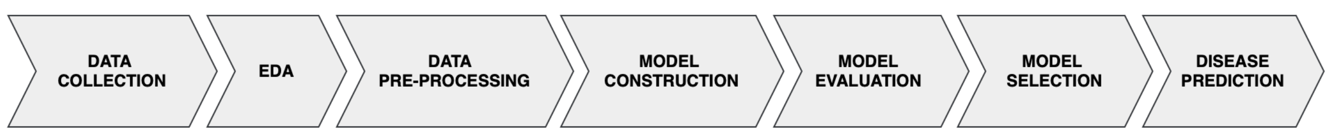
In response to these challenges, our project aims to leverage advanced technology to enhance the accessibility and efficiency of eye care. By developing a sophisticated classification system that utilizes machine learning and deep learning techniques, we seek to provide rapid and accurate diagnoses of common eye diseases such as Diabetic Retinopathy, Cataract, and Glaucoma through the analysis of retinal images. This initiative promises to deliver instant feedback to healthcare professionals and patients, particularly in remote regions where access to specialized care is limited.

This report outlines the objectives, methodologies, and anticipated impact of our project, highlighting how innovative technological solutions can transform eye care delivery and ultimately improve health outcomes for vulnerable populations.

Goal

Retinanet is web-based eye disease detection application which will integrate the best machine learning architecture out of the three approaches done in the project

Flow Chart



Data

4 datasets are being used in this project.

- The first dataset is the Ocular Disease Recognition (ODIR-5K) dataset. It allows the model to classify the image as normal or the presence of a disease such as DR, Cataract, Glaucoma, or other ocular conditions.
- The second dataset in use is the Cataract dataset which allows the model to classify images into no-cataract and cataract-detected.
- The third dataset is the Diabetic retinopathy dataset which allows the model to classify images into no DR, mild DR, moderate DR, severe DR, and proliferative DR.

- The fourth dataset in use is the Glaucoma Detection Dataset which allows the model to classify images into no glaucoma and glaucoma detected.

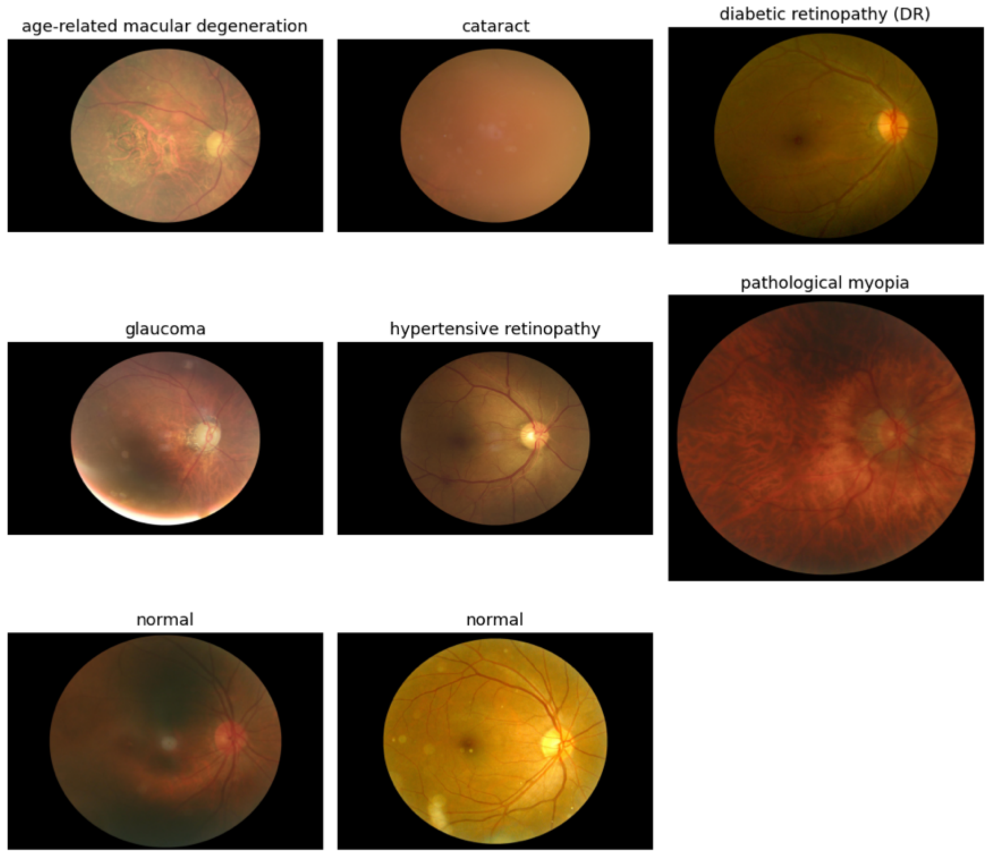
ODIR-5K

The ODIR-5K dataset is a structured ophthalmic database, comprising records of 5,000 patients, developed to represent patient information. The dataset includes a csv file and image set. The CSV file consists of the ID, left-fundus and right-fundus, age, gender, diagnosis, left-diagnostic keywords, and right-diagnostic keywords as columns. The dataset consists of 5000 patients, with left-eye and right-eye images. The ODIR-5k dataset has eight output labels for classifying ocular diseases. These labels are:

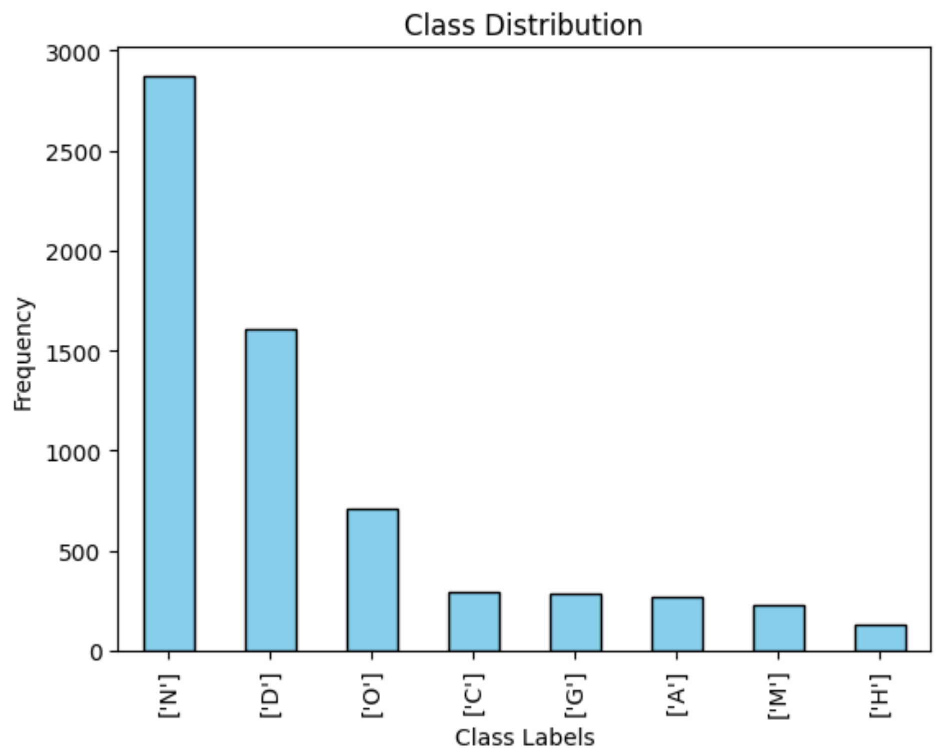
a. Normal (N) - 4490
b. Diabetic Retinopathy (D) - 2520
c. Glaucoma (G) - 1110
d. Cataract (C) - 460
e. Age-related Macular Degeneration (A) - 440
f. Hypertension (H) - 420
g. Myopia (M) - 360
h. Other Diseases (O) - 200

EDA

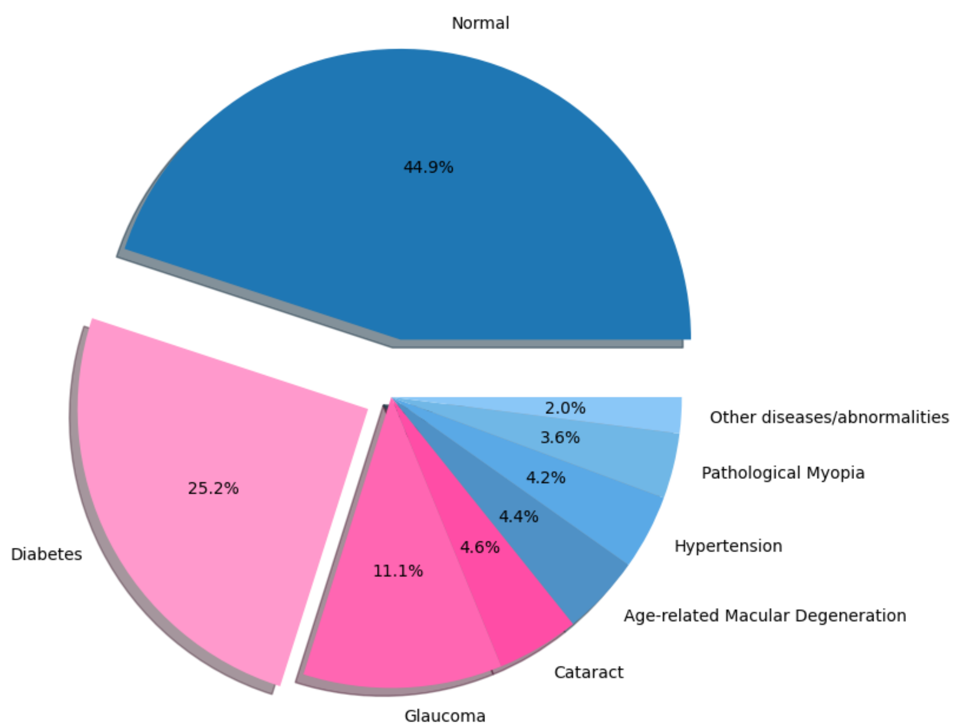
A. Visualization



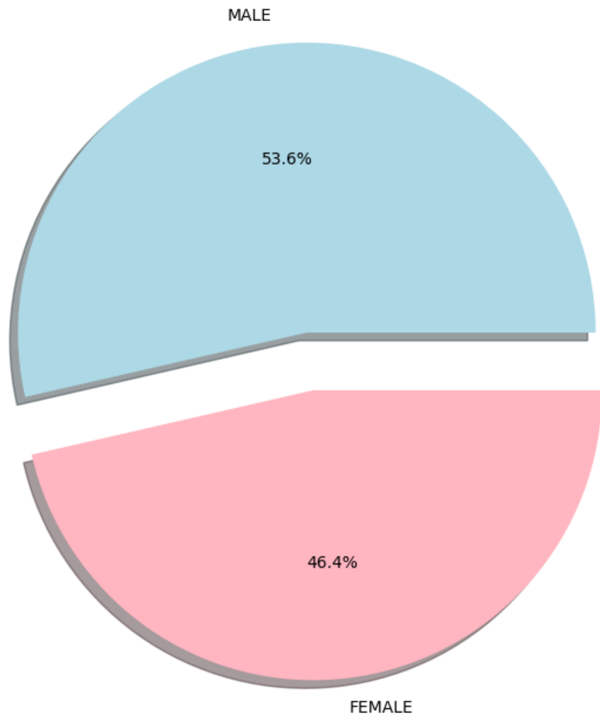
B. Class distribution



C. Class distribution – pie chart



D. Gender distribution



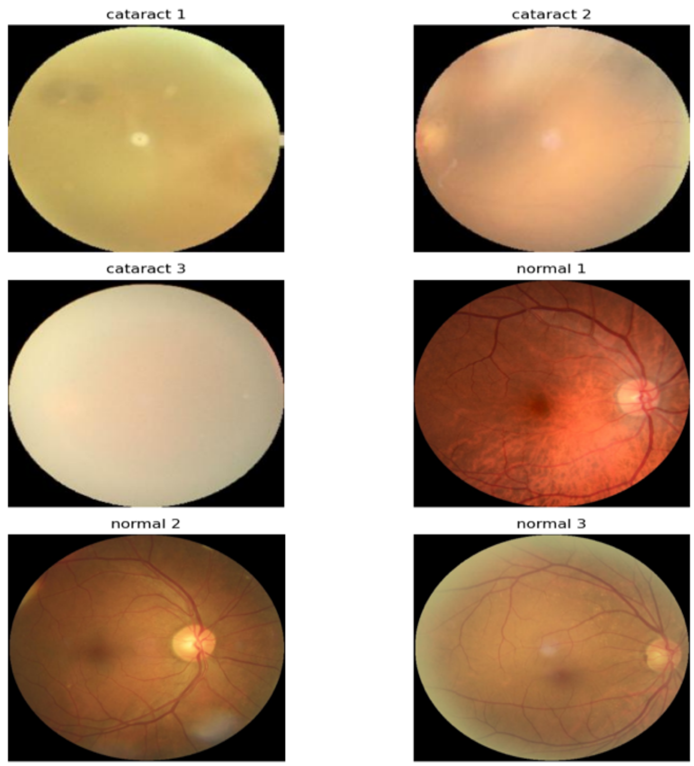
Cataract

The Cataract Classifier Dataset is a specialized collection of eye images, labeled to indicate the presence or absence of cataracts. This dataset serves as a focused resource for training models in binary classification of cataract conditions, differentiating between cataract-affected and normal eyes. The dataset is structured with two main classes:

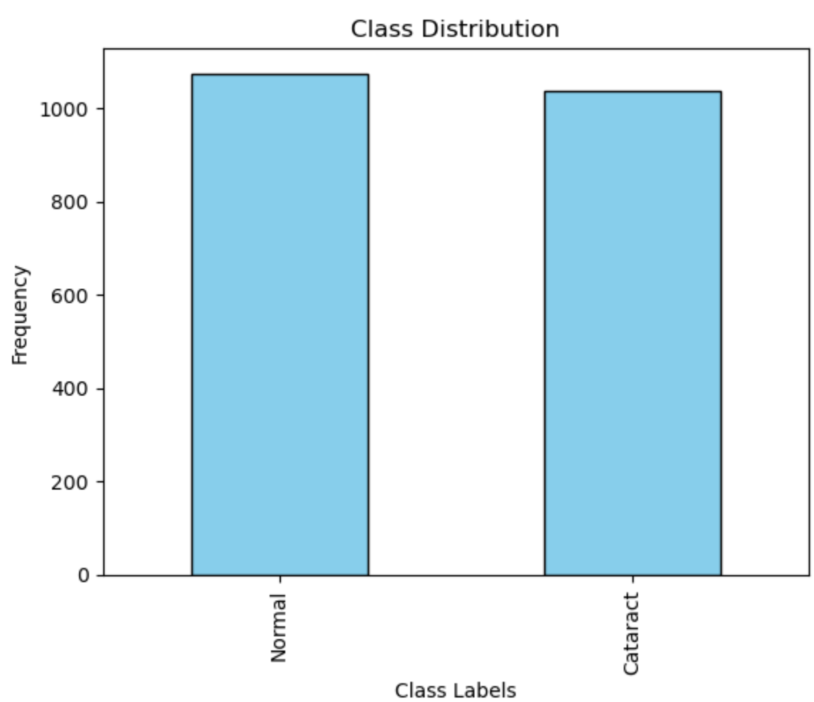
- Normal: Images of eyes without cataracts, totaling 300 images.
- Cataract: Images of eyes with visible cataracts, totaling 100 images

EDA

A. Visualization



B. Class Distribution



Diabetes Retinopathy (DR)

In this dataset, resized version of the dataset, and a cropped then resized version of the data are both included.

trainLabels.csv:

This file contains the name of the file under the 'image' column and the label under the 'level' column. The size of this csv file is 465.32 kB, and the number of the records is 35126. Unique values in the ‘level’ are (0,1,2,3,4) each representing the stages of the diabetes retinopathy.

level	Stages	number of images
0	Normal	25802
1	Mild DR	2438
2	Moderate DR	5288
3	Severe DR	872
4	Proliferative DR	708

resized_train:

This folder was created by simply resizing the dataset to 1024x1024 if it is bigger than this size, else it remains the same.

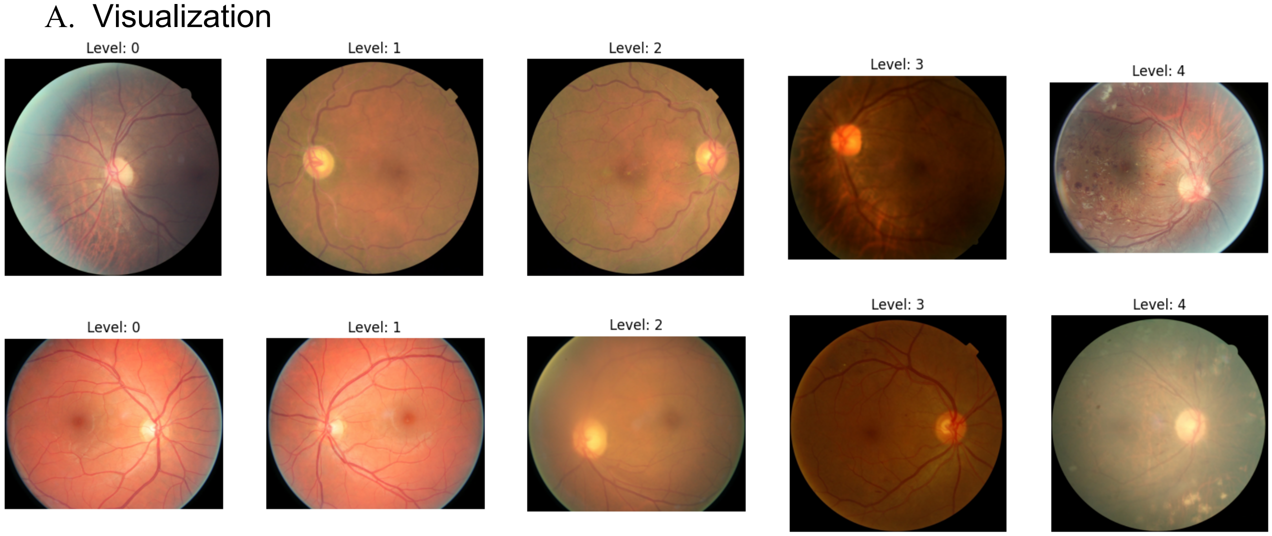
resized_train_cropped:

In this case, black space is cropped out by trying to identify the center and radius of the circle of the fundus image. Some of the images turned out to be fully black or very close to fully black, and no mask was found. Hence, those images were manually removed.

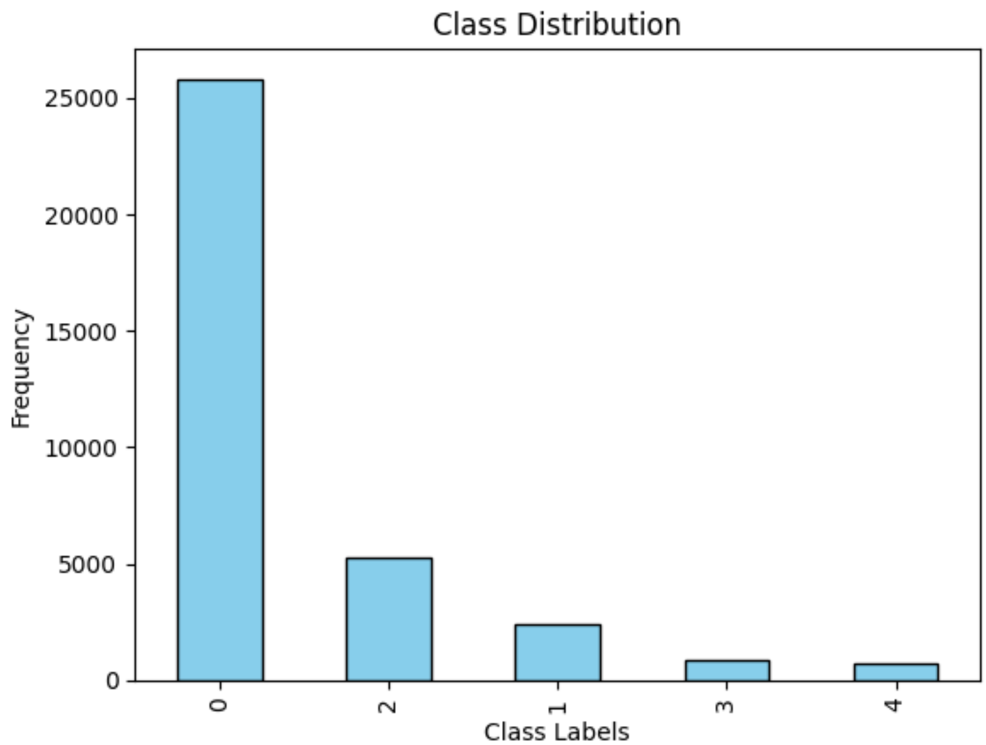
Level 0 has more images than the other levels. But under sampling the dataset decreases the accuracy of the model.

EDA

A. Visualization



B. Class Distribution



Glaucoma

The glaucoma data set contains images/oct scans of the eye in the .jpg format. Data set size is 5 columns x 650 rows. Each image is standardized to 224x224x3.

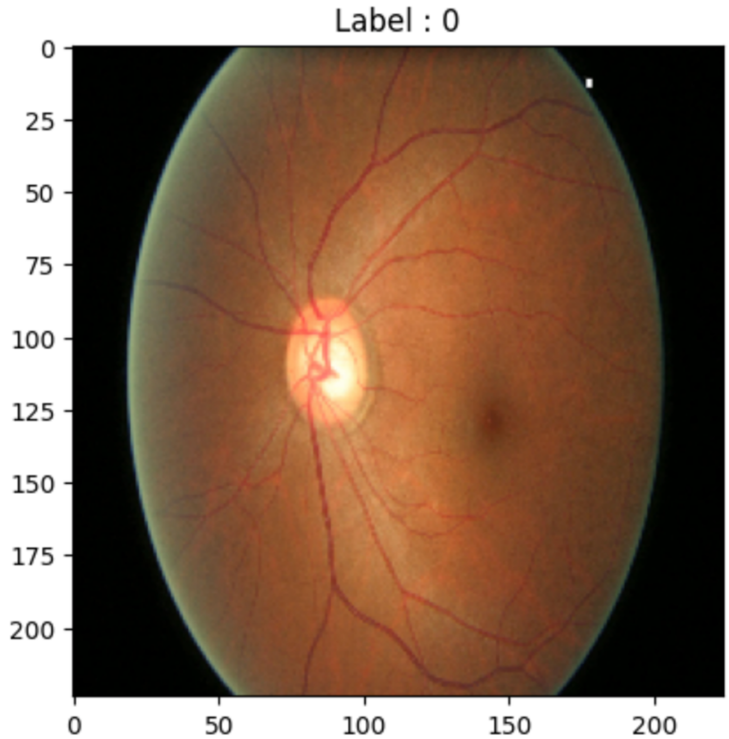
LABEL	NO OF SAMPLES
With Glaucoma	168
Without Glaucoma	482

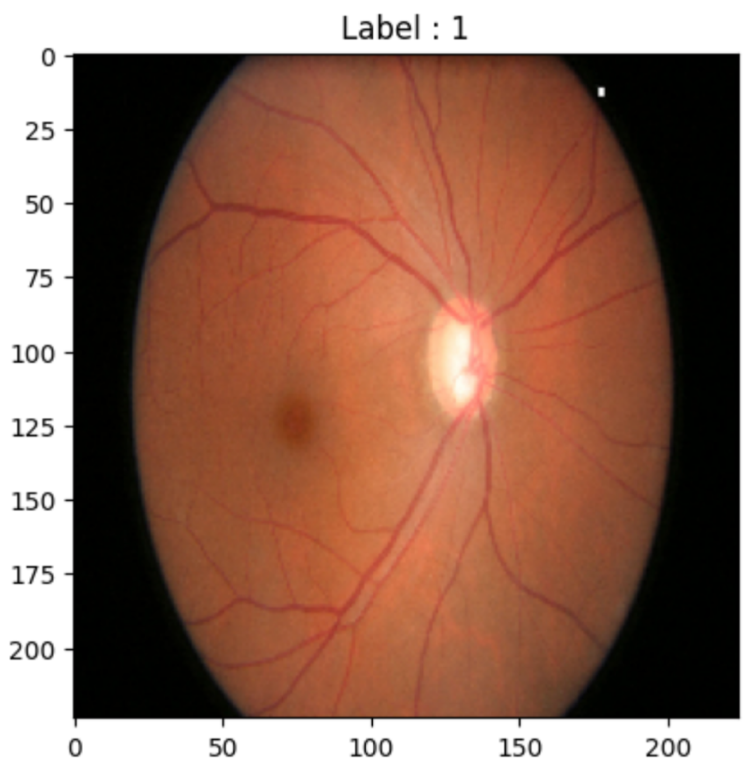
From the firsthand analysis, we can see that the dataset is heavily imbalanced. This problem is addressed by using class weights api in sklearn utils, which helps in the training phase itself. This is ideal for small datasets where chances of overfitting is high.

After pre-processing the images, mapping between images and the csv file is done. Then using the batch wise image processing tool – ImageDataGenerator in tensorflow, train image generator and test image generator is created. The split ratio is 0.2.

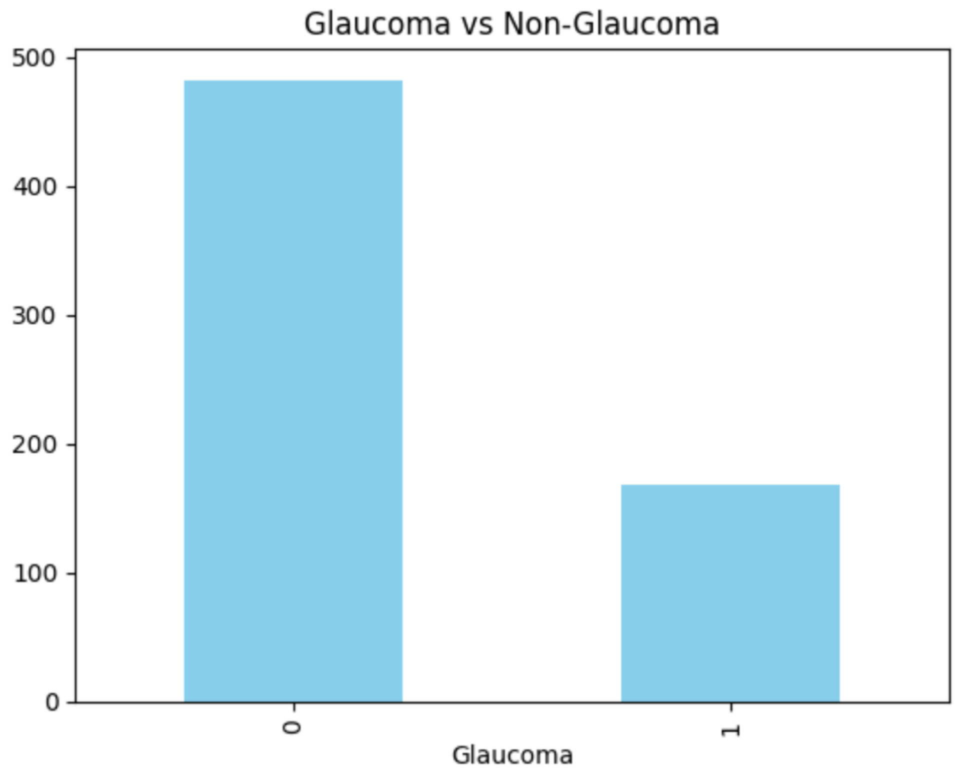
EDA

A. Visualization



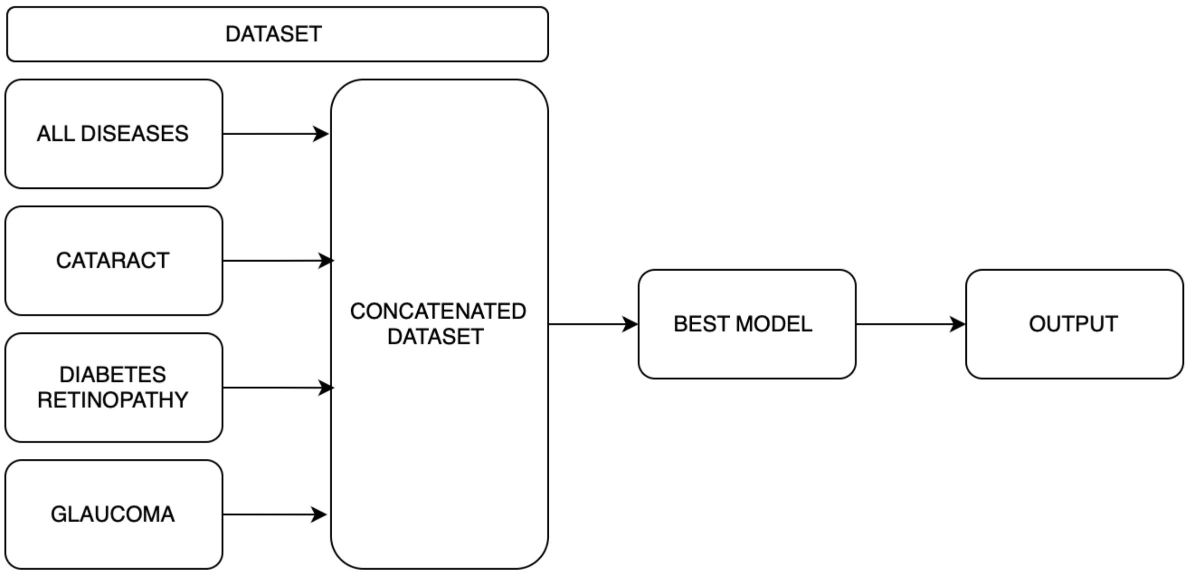


B. Class Distribution



Different architectures and approaches

Version 1



Data preparation

The initial version of the model is a single-class classification model trained on a comprehensive dataset that combines all the aforementioned sources. This unified dataset has been standardized and organized to include three key columns:

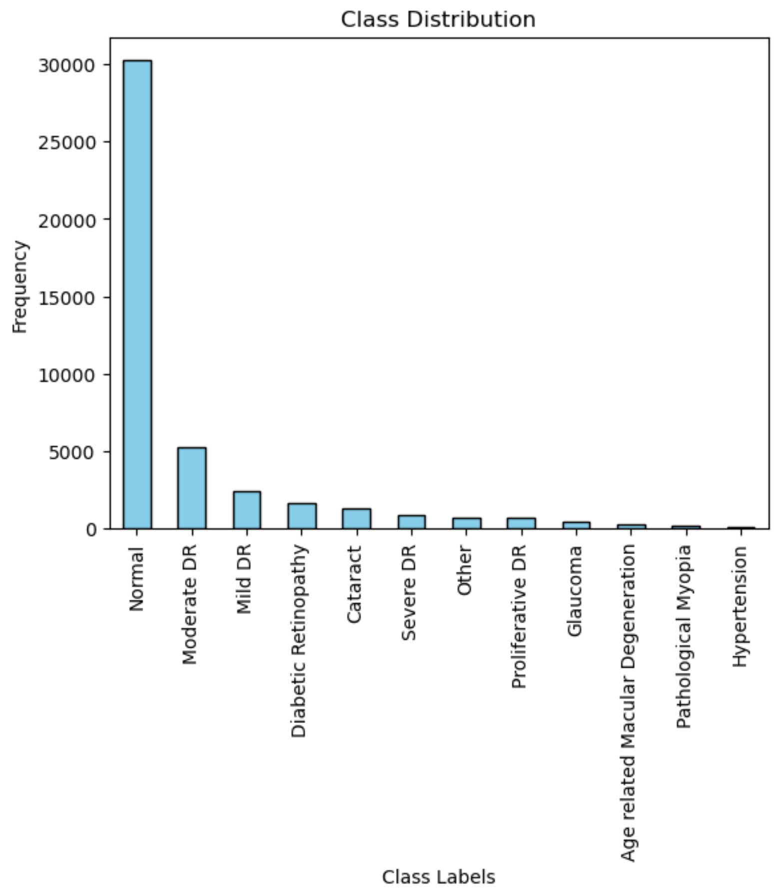
- image_path - specifies the location of each image
- label – indicates the diagnosed disease
- source - identifies the original dataset

By integrating data from multiple sources, the model benefits from a diverse array of examples, ensuring a more robust training foundation for detecting various eye diseases.

	image_path	label	source
0	classifier_one_data/../input/ocular-disease-re...	Normal	classifier_one
1	classifier_one_data/../input/ocular-disease-re...	Normal	classifier_one
2	classifier_one_data/../input/ocular-disease-re...	Diabetic Retinopathy	classifier_one
3	classifier_one_data/../input/ocular-disease-re...	Diabetic Retinopathy	classifier_one
4	classifier_one_data/../input/ocular-disease-re...	Diabetic Retinopathy	classifier_one
...	...	...	...
44257	glaucoma_data/archive/ORIGA/ORIGA/Images/646.jpg	Glaucoma	glaucoma
44258	glaucoma_data/archive/ORIGA/ORIGA/Images/647.jpg	Glaucoma	glaucoma
44259	glaucoma_data/archive/ORIGA/ORIGA/Images/648.jpg	Glaucoma	glaucoma
44260	glaucoma_data/archive/ORIGA/ORIGA/Images/649.jpg	Normal	glaucoma
44261	glaucoma_data/archive/ORIGA/ORIGA/Images/650.jpg	Glaucoma	glaucoma

44262 rows × 3 columns

1. Class Distribution



Models used

Base classifier models: MobileNetV2, Xception
Chosen model: Both gave the same accuracy

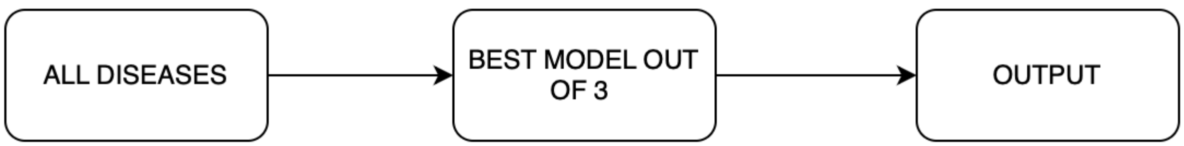
1. MobileNetV2 leverages an inverted residual structure and linear bottlenecks, providing a balance between performance and computational efficiency. It performs exceptionally well on diverse datasets by preserving the spatial resolution of medical images while minimizing the number of operations. Starting with a pre-trained ImageNet version of MobileNetV2, the model is fine-tuned for our classification task by adding fully connected layers, enabling it to focus on disease-specific features and labels.
2. Xception utilizes depth wise separable convolutions, significantly lowering computational costs while retaining accuracy, making it ideal for handling large, diverse datasets. It excels in feature extraction by separating spatial and channel information, which is particularly beneficial for analyzing detailed medical images. The model starts with a pre-trained Xception base (excluding the top layers) and incorporates fully connected layers for classification, allowing it to specialize in predicting specific disease labels.

Version 2

Separate models are built for predicting the disease.

Models	Dataset
Model 1	ODIR-5K
Model 2	Cataract
Model 3	Diabetes Retinopathy
Model 4	Glaucoma

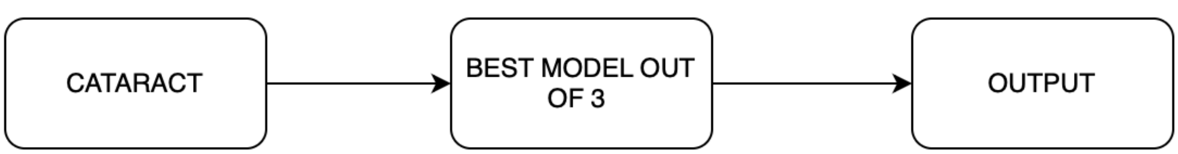
Model 1: Base classifier



Base classifier models: EfficientNetB7, Xception, Custom CNN
Chosen model: Xception

1. EfficientNetB7 is chosen for its high accuracy and efficiency in image classification, using compound scaling to balance model depth, width, and resolution. It is computationally efficient, making it suitable for high-resolution medical imaging tasks. The EfficientNetB7 base model is used with pre-trained ImageNet weights and customized with additional dense layers for classification, followed by a softmax layer to output probabilities for each class.
2. Xception makes use of depthwise separable convolutions, reducing computational cost while maintaining accuracy, ideal for large and diverse datasets. It offers a high level of feature extraction by decoupling spatial and channel information, beneficial for detailed medical images. Pre-trained Xception is applied as the base model with top layers removed, and fully connected layers are added for classification to adapt the model to specific target labels.
3. The custom cnn model allows for flexibility in architecture, allowing custom layers and configurations tailored to the dataset and classification task. It offers control over feature extraction, enabling adjustments based on dataset characteristics. It consists of multiple convolutional, pooling, and dense layers, with dropout for regularization, culminating in a softmax layer to classify multiple categories accurately.

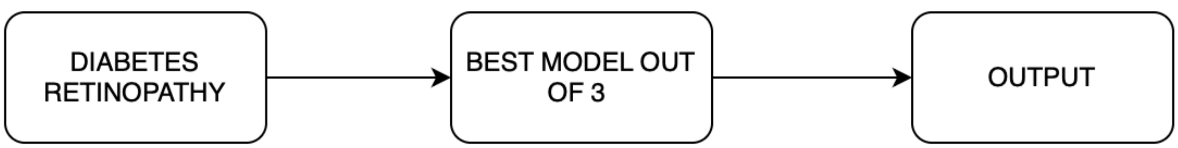
Model 2: Cataract classifier



Cataract classifier models: EfficientNetB7, MobileNetV2, Custom CNN + RNN
Chosen model: MobileNetV2

1. EfficientNetB7 is highly effective for processing high-resolution medical images, helping achieve optimal performance with minimal computational cost. Its balanced scaling approach allows it to handle subtle features in cataract images, important for identifying slight variations indicative of disease. EfficientNetB7 is used as a frozen feature extractor with ImageNet weights, and additional dense layers fine-tune the model for binary classification between cataract and normal images.
2. MobileNetV2 is lightweight and optimized for mobile devices, ideal for real-time medical diagnostics in resource-constrained environments. It is computationally effective. It is configured with pre-trained weights and additional layers for classification, making it adaptable for classifying cataract and normal eye images with a softmax output layer
3. Custom CNN + RNN enables the model to capture spatial and sequential features, useful for analyzing medical image sequences. This can capture temporal dependencies in eye scans, potentially enhancing classification performance for progressive diseases. The architecture consists of several convolutional layers for feature extraction, followed by an RNN layer to capture sequential patterns, with a softmax layer to differentiate cataracts from normal image

Model 3: Diabetes Retinopathy classifier

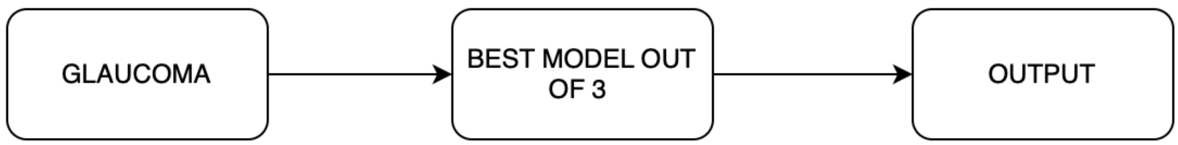


DR classifier models: EfficientNetB7, MobileNetV2, Xception
Chosen model: Xception

1. EfficientNetB7 is exceptionally suited for high-resolution medical imaging, offering strong performance with minimal computational cost. Its balanced scaling efficiently captures subtle features in cataract images, which is crucial for identifying minute variations that signal disease presence. Leveraging EfficientNetB7 as a frozen feature extractor with pre-trained ImageNet weights, additional dense layers are added to fine-tune the model specifically for binary classification between cataract and normal images.
2. MobileNetV2 leverages an inverted residual structure and linear bottlenecks, providing a balance between performance and computational efficiency. It performs exceptionally well on diverse datasets by preserving the spatial resolution of medical images while minimizing the number of operations. Starting with a pre-trained ImageNet version of MobileNetV2, the model is fine-tuned for our classification task by adding fully connected layers, enabling it to focus on disease-specific features and labels.

3. Xception utilizes depth wise separable convolutions, significantly lowering computational costs while retaining accuracy, making it ideal for handling large, diverse datasets. It excels in feature extraction by separating spatial and channel information, which is particularly beneficial for analyzing detailed medical images. The model starts with a pre-trained Xception base (excluding the top layers) and incorporates fully connected layers for classification, allowing it to specialize in predicting specific disease labels.

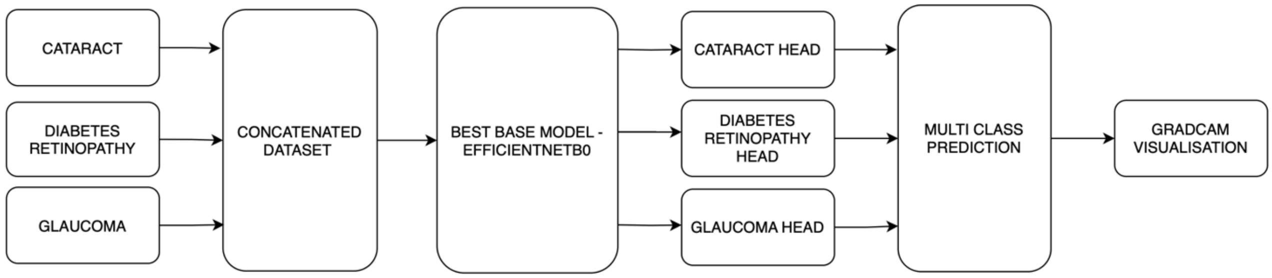
Model 4: Glaucoma classifier



Models used: EfficientNetB7, MobileNetV3Small, ResNet50
Chosen Model: EfficientNetB7

1. EfficientNetB7 is a powerful variant in the EfficientNet family, optimized for balancing model accuracy and computational efficiency through a compound scaling method that adjusts depth, width, and resolution. This architecture is especially suitable for medical image analysis due to its ability to achieve high accuracy with relatively efficient processing, making it effective for handling the high resolution often required in medical imaging. By using EfficientNetB7 as a base and modifying the output layers, we can tailor the model to classify medical conditions with strong precision and reduced computational overhead.
2. MobileNetV3Small is a lightweight neural network architecture that prioritizes speed and efficiency, originally designed for mobile applications. This model is ideal for medical imaging tasks where quick inference is critical, such as in real-time diagnostics on portable devices. Its compact architecture and depth wise separable convolutions make it an effective base model when modified with specialized output layers for medical image classification, balancing accuracy and speed for rapid and resource-efficient analysis.
3. ResNet50 is a deep residual network with 50 layers, known for its use of skip connections that address vanishing gradient issues, allowing for deeper architectures and better feature extraction. This capability is particularly valuable in medical imaging, where complex patterns must be learned from high-dimensional data. As a base model with modified output layers, ResNet50 can capture intricate details in medical images, making it well-suited for tasks like disease detection and classification where high accuracy and deep feature understanding are essential

Version 3



This is an integrated deep-learning model for diagnosing multiple ophthalmic diseases using fundus images. Specifically, the task is to identify the presence of cataracts, glaucoma, and diabetic retinopathy (DR) from a given dataset. Each disease represents a different classification task: cataract and glaucoma are binary classification problems

The model utilizes EfficientNetB0 as the base architecture. EfficientNet is a convolutional neural network that is highly efficient in balancing accuracy and computational complexity. Its use here ensures high performance while being less resource intensive.

Base Model: The base feature extractor is the EfficientNetB0 model, pre-trained on the ImageNet dataset. Using a pre-trained model helps transfer useful visual features from a general dataset to the specific problem of fundus image classification. Transfer learning is beneficial as it allows faster convergence and better accuracy, especially with smaller datasets.

Task-Specific Heads: After extracting features using the EfficientNet base, three separate output heads are added for each classification task:
Cataract Output Head: A fully connected layer with a sigmoid activation function for binary classification of cataracts.
Glaucoma Output Head: Another binary classification head for glaucoma.
Diabetic Retinopathy Output Head: A multi-class output layer with softmax activation for classifying diabetic retinopathy severity.

These separate heads allow the network to learn multiple tasks simultaneously, leveraging shared features while optimizing each task individually.

<u>Significance of the integrated model</u>
Advantages of integrated model:

1. Shared Learning: Multiple diseases can be identified using the same set of image features. By sharing the underlying network for different disease classifications, the model benefits from feature reuse, improving generalization, and reducing overfitting.
2. Efficiency: Instead of training three separate models, this approach integrates them, saving computational resources and providing a more unified diagnostic tool.

3. Clinical Application: In practice, a single patient may need to be screened for multiple conditions. A multi-output model is better suited for such use cases, where multiple diagnoses can be made from a single fundus image in one inference step.

Results and analysis

Version 1 architecture

Model	accuracy	loss
MobileNet2	65.43	155.7598
Xception	66.02	139.6008

Version 2 architecture

Base classifier

Model	accuracy	loss
EfficientNetB7	83.99	0.3873
Xception	86.76	0.2170
Simple CNN	51.35	1.2199

Cataract classifier

Model used	accuracy	loss
EfficientNetB7	50.98	0.6934
MobileNetV2	98.08	0.0453
Custom CNN + RNN	73.12	0.5147

Diabetes Retinopathy classifier

Model used	accuracy	loss
EfficientNetB7	73.31	0.9647
MobileNetV2	73.19	0.8777
Xception	87.78	0.2668

Glaucoma classifier

Model used	accuracy	loss
EfficientNetB7	73.19	0.6268
MobileNetV3Small	72.33	0.6644
ResNet50	26.32	0.7270

Version 3 architecture

Model used	accuracy
EfficientNetB0 - cataract	99.44
EfficientNetB0 - dr	77.2
EfficientNetB0 - glaucoma	99.6

Conclusion

Based on the Result and Analysis, out of the 3 versions, Version-3 architecture gives the most efficient and interpretable results. So, Version-3 is deployed as the final classifier for Eye disease detection.
Implementation is done using Grad-CAM and Flask as described below.

Grad-CAM implementation

Grad-CAM (Gradient-weighted Class Activation Mapping) is implemented to provide visual explanations for the model's predictions. Grad-CAM is a crucial tool for understanding the regions in the input image that contribute the most to the model's decision-making process. The Grad-CAM implementation involves extracting the gradients of the target output concerning the last convolutional layer's output. These gradients are then used to compute weights, which are applied to the feature maps to generate a heatmap. The heatmap is then superimposed on the original image to visually highlight the regions that contributed most to the prediction. This implementation can be applied to each of the three classification tasks (cataract, glaucoma, and DR), thereby generating interpretability for each type of disease prediction.
In medical image analysis, explainability is vital for building trust in the AI model, especially for healthcare professionals. Using Grad-CAM allows ophthalmologists to see which areas of the retinal image contributed to a specific prediction, providing transparency and aiding in validating the model's correctness. This interpretability is especially important when deploying AI in critical decision-making scenarios like disease diagnosis.

Flask implementation

A web interface was developed using Flask, HTML, and CSS, to create an accessible interface for diagnosing eye conditions such as cataracts, glaucoma, and diabetic retinopathy (DR) using a machine learning model. Users can upload images of eyes, receive predictions on potential conditions, and view Grad-CAM visualizations to understand which areas of the image influenced the diagnosis.

Technologies Used

1. Flask: A lightweight Python web framework used for rendering HTML templates.

2. HTML & CSS: Used to create the frontend structure and style of the web interface.
3. OpenCV: Used for image processing tasks, particularly for creating Grad-CAM visualizations

File Structure and Purpose

1. app.py: Main Flask application file that sets up routes for different pages, handles user uploads, and coordinates between user interactions and backend processing.
2. static/styles.css: Defines the web interface's color theme and layout, ensuring a cohesive and visually appealing design across pages.
3. templates/: Contains HTML templates for different pages such as home.html, input.html, output.html, and gradcam.html, each providing unique functionality.
4. model/integrated_model.py: Loads the integrated model, defining its structure and loading weights for the eye condition detection tasks.
5. model/predict.py: Contains functions for image preprocessing, predicting conditions, and generating Grad-CAM visualizations.
6. samples/: A directory for storing user-uploaded images and Grad-CAM output images.

Page Descriptions and Functions

page	Function	Utility
home.html	Introduces the application and directs users to the image upload page.	Acts as the entry point for users, creating a welcoming and informative start to the diagnostic process.
input.html	A form that accepts image uploads, which are then saved in the samples/ directory.	Collects user data for the diagnostic process and redirects to the output page after successful upload
output.html	Displays the model's predictions, including probabilities of cataracts, glaucoma, and DR classification.	Provides feedback to the user on the model's assessment, summarizing the diagnostic results.
gradcam.html	Shows Grad-CAM output, highlighting areas of the uploaded image that the model used for its classification	Offers transparency to users by visually explaining how the model made its diagnosis, which enhances interpretability

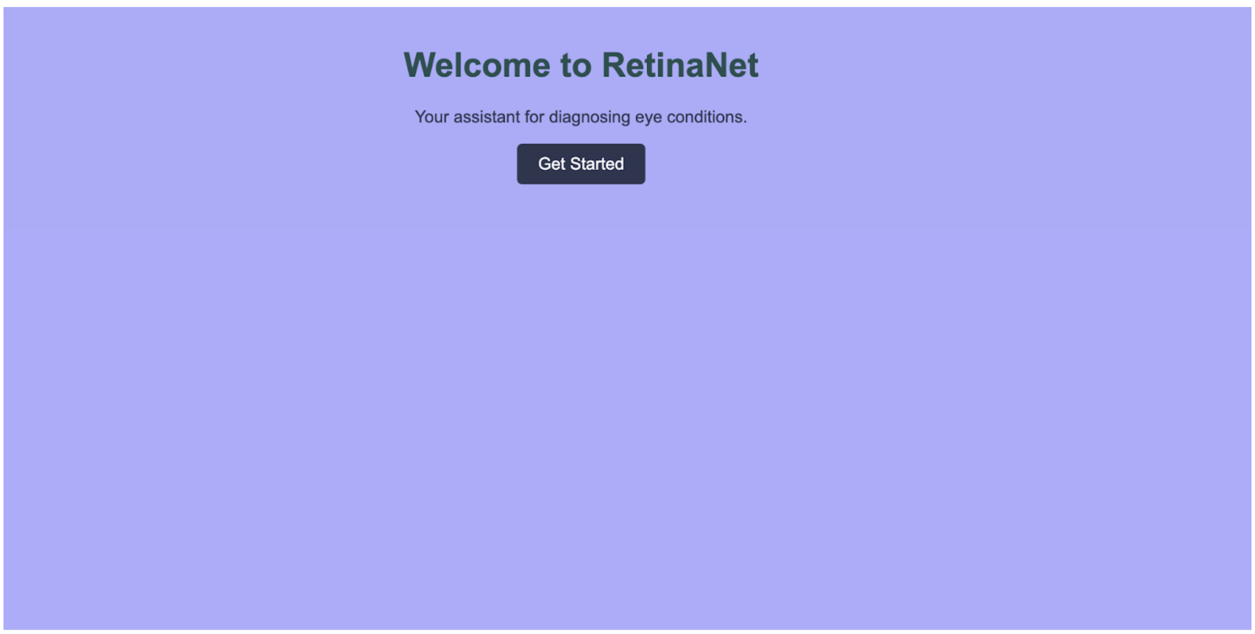
Models used

1. **Model Integration and Loading** (integrated_model.py):

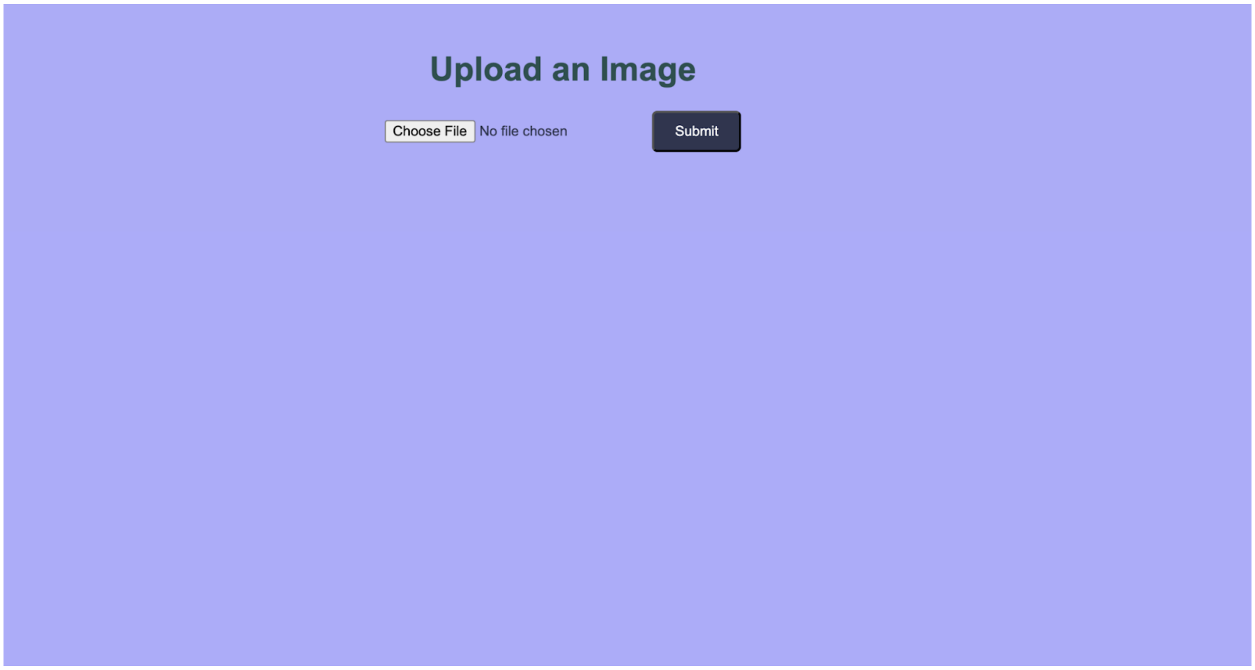
The file defines and loads a pre-trained model architecture specifically designed for multi-task learning. The model's weights are loaded from an h5 file, ensuring it's ready for inference when the server is running.

2. **Image Prediction and Grad-CAM Generation** (predict.py):
    a. Image Preprocessing and Prediction: process_image() handles the image processing. It resizes, scales, and normalizes the image, passing it through the model to obtain probabilities for cataracts and glaucoma and a classification for diabetic retinopathy.
    b. Grad-CAM Visualization: generate_gradcam() creates a Grad-CAM image by computing gradients concerning the target class. Using OpenCV, the Grad-CAM heatmap is overlaid on the original image to produce a final visualization, saved in the uploads/ directory for display on the Grad-CAM page.

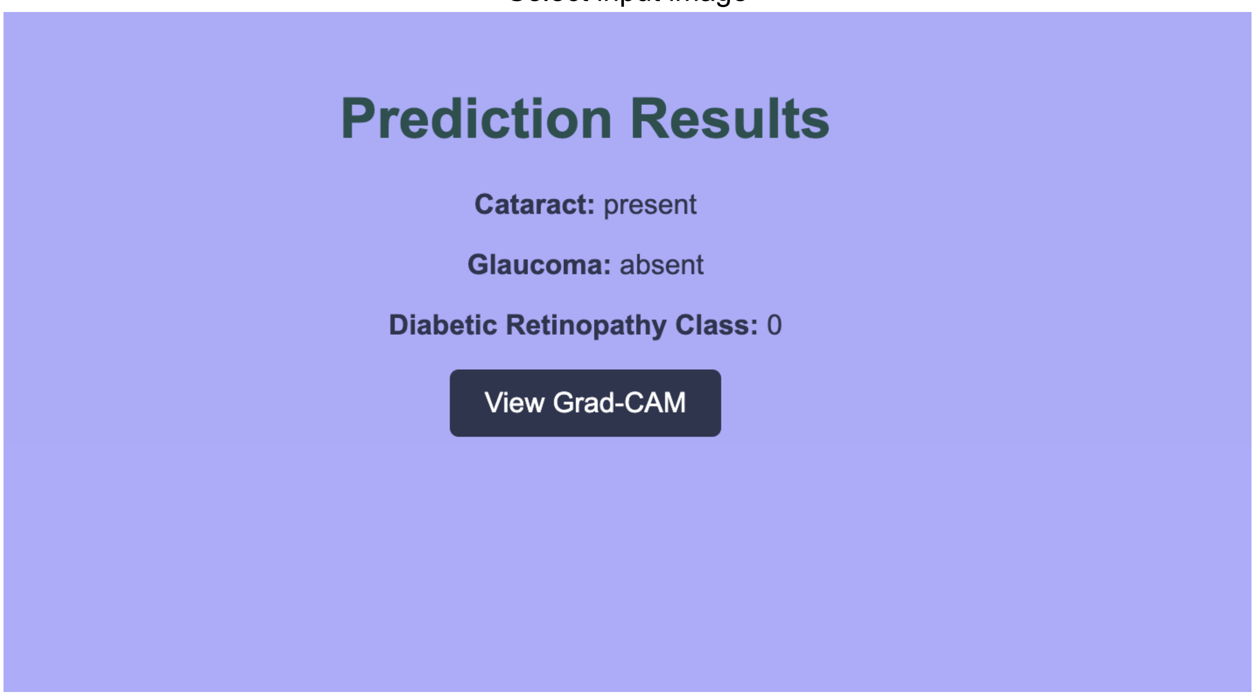
Images from web interface



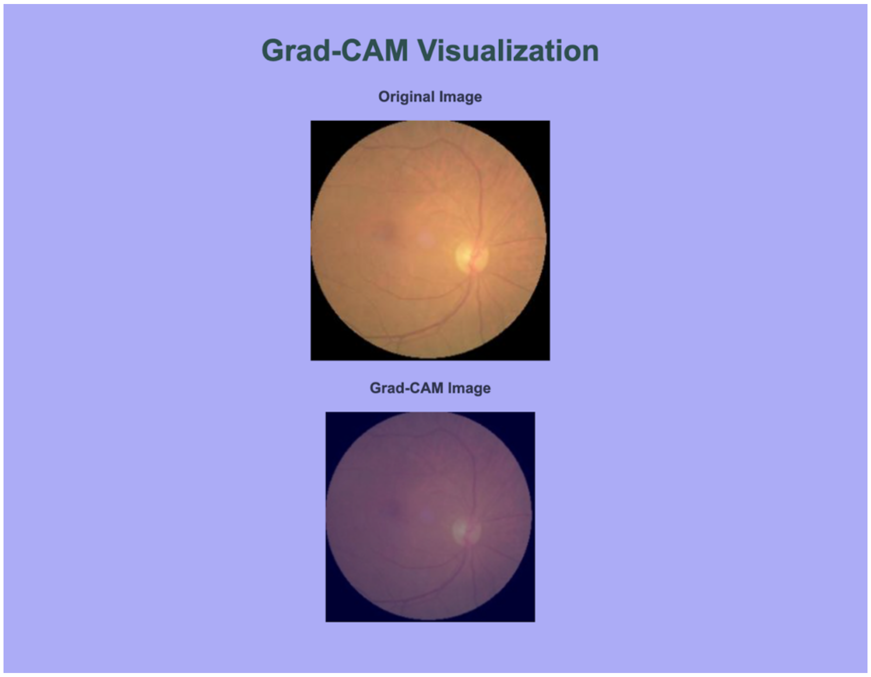
Home page



Select input image



Predict results



Gradcam implementation

Relevance with the PRS module

Retinanet is a pattern recognition for the medical image data, especially in the field of ophthalmology. Using RetinaNet with transfer learning greatly enhances a pattern recognition system for retinal disease detection in ophthalmology. Transfer learning enables RetinaNet to leverage pretrained models, achieving high accuracy and faster convergence even with limited labeled medical data. This approach improves feature representation for subtle disease markers, minimizes computational costs, and allows for better generalization across various imaging techniques. The combination optimizes RetinaNet for real-time, scalable disease detection, supporting clinicians in early and accurate diagnoses to prevent vision loss.