**Eye Disease Classification Using Deep Learning and Flask**

This project focuses on **automated classification of eye diseases** using a **deep learning model** and a **Flask-based web application**. The dataset, sourced from Kaggle, contains images categorized into **Cataract, Diabetic Retinopathy, Glaucoma, and Normal**. The images are preprocessed using **ImageDataGenerator** to standardize size, normalize pixel values, and apply augmentation techniques like rotation, zooming, and flipping to enhance model robustness. The model is built using **Transfer Learning** with **ResNet50**, a pre-trained convolutional neural network (CNN), where the base layers are frozen, and custom **fully connected layers** are added for classification. The model is compiled using the **Adam optimizer** with categorical cross-entropy loss and trained for multiple epochs using a **training-validation split**. After training, the model is saved as eye\_disease\_model.h5 for deployment.

A **Flask web application** is developed to allow users to **upload eye images** and receive real-time predictions. The backend loads the trained model, preprocesses the uploaded image, and predicts the disease class. The UI, designed with **HTML and CSS**, features a simple upload interface (index.html) and a result display page (result.html). Uploaded images are stored in a static folder, and predictions are displayed with the corresponding **eye disease name**. The web app is executed using app.py, and users can access it via http://127.0.0.1:5000/ in a browser. The project is structured with separate folders for **static files, templates, model storage, and training scripts**, ensuring a modular and organized workflow.

This project demonstrates **the power of AI in healthcare**, offering a **quick and efficient** method to detect **common eye diseases** using **computer vision**. Future improvements can include **fine-tuning the model with more layers**, **expanding the dataset**, and **deploying the application on a cloud platform** like **Heroku or AWS** for wider accessibility. 🚀