**Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy**

**Project Report**

1. **INTRODUCTION**
   1. **Project Overview:**

[Diabetic Retinopathy](https://www.sciencedirect.com/topics/medicine-and-dentistry/diabetic-retinopathy) (DR) is a common [complication of diabetes mellitus](https://www.sciencedirect.com/topics/medicine-and-dentistry/complications-of-diabetes-mellitus), which causes lesions on the retina that effect vision. If it is not detected early, it can lead to [blindness](https://www.sciencedirect.com/topics/medicine-and-dentistry/amaurosis). Unfortunately, DR is not a reversible process, and [treatment](https://www.sciencedirect.com/topics/medicine-and-dentistry/therapeutic-procedure) only sustains vision. DR early detection and treatment can significantly reduce the risk of [vision loss](https://www.sciencedirect.com/topics/medicine-and-dentistry/visual-impairment).

The project aims to develop an advanced system for early detection of diabetic retinopathy through deep learning-based analysis of fundus images.

* 1. **Purpose**

Diabetic [retinopathy](https://www.sciencedirect.com/topics/medicine-and-dentistry/retinopathy) (DR) is one of the leading causes of preventable [blindness](https://www.sciencedirect.com/topics/medicine-and-dentistry/amaurosis) globally. Performing retinal screening examinations on all diabetic patients is an unmet need, and there are many undiagnosed and untreated cases of DR. The objective of this study was to develop robust diagnostic technology to automate DR screening. Referral of eyes with DR to an ophthalmologist for further evaluation and [treatment](https://www.sciencedirect.com/topics/medicine-and-dentistry/therapeutic-procedure) would aid in reducing the rate of [vision loss](https://www.sciencedirect.com/topics/medicine-and-dentistry/visual-impairment), enabling timely and accurate diagnoses.

1. **LITERATURE SURVEY**
   1. **Existing problem**

Diabetic retinopathy is a leading cause of blindness, and early detection is crucial for effective treatment. Current methods are time-consuming and may lack accuracy.

* 1. **References**

<https://www.sciencedirect.com/science/article/abs/pii/S0161642016317742>

<https://www.sciencedirect.com/science/article/pii/S2352914820302069>

<https://ieeexplore.ieee.org/abstract/document/8869883>

* 1. Problem Statement Definition

The lack of accessible and timely diabetic retinopathy diagnosis poses a lesion on the retina that affect vision to patients with diabetes. Developing a precise, efficient, and cost-effective deep learning-based fundus image analysis system is imperative for early detection, ensuring timely intervention and preventing irreversible vision loss.

1. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas

Mural link:

<https://app.mural.co/t/aiml1520/m/aiml1520/1697207642142/5bf7f4e75c3e4ff0e027bd457a50ac4e7441c0e5?sender=uc481eff1f1018327189c3823>

A screenshot of a computer

Description automatically generated

* 1. **Ideation & Brainstorming**

**Ideation Phase**

**Brainstorm & Idea Prioritization Template**

|  |  |
| --- | --- |
| Date | 18 October 2023 |
| Team ID | Team-593212 |
| Project Name | Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy |
| Maximum Marks | 4 Marks |

**Brainstorm & Idea Prioritization Template:**

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference:

<https://app.mural.co/t/aiml1520/m/aiml1520/1697433256698/9bd58e57352a146f4b255aaf5faa10b2141f6b3c?sender=uc481eff1f1018327189c3823>

**Step-1: Team Gathering, Collaboration and Select the Problem Statement**

A screenshot of a computer

Description automatically generated

**Step-2: Brainstorm, Idea Listing and Grouping**

A screenshot of a computer

Description automatically generated

**Step-3: Idea Prioritization**

A screen shot of a diagram

Description automatically generated

1. **REQUIREMENT ANALYSIS**
   1. **Functional requirement**

* Fundus image input processing
* Deep learning model for diabetic retinopathy classification
* User interface for results interpretation
  1. **Non-Functional requirements**
* System reliability
* Scalability for future enhancements
* User-friendly interface

1. **PROJECT DESIGN**
   1. **Data Flow Diagrams & User Stories**

**Project Design Phase-II**

**Data Flow Diagram & User Stories**

|  |  |
| --- | --- |
| Date | 22 October 2023 |
| Team ID | Team-593212 |
| Project Name | Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy |
| Maximum Marks | 4 Marks |

**Data Flow Diagrams:**

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



**User Stories:**

The below table is the list for all the user stories for the product.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (Web user) | Access | USN-1 | As a user, I can enter to the DRetina Check Website. | I can access the DRetina Check Web page | High | Sprint-1 |
|  |  | USN-2 | As a user, I can use the interactive DRetina Check Website. | I can see the DRetina Check Web components | High | Sprint-1 |
|  |  | USN-3 | As a user, I can choose the images from my device for the prediction. | I can access the image to choose from the device | High | Sprint-1 |
|  |  | USN-4 | As a user, I can see the preview of the chosen image. | I can access the preview image | High | Sprint-1 |
|  |  | USN-5 | As a user, I can see the classification of the image and the medication suggestions | I can see the identified class and suggestions | High | Sprint-1 |
|  |  | USN-6 | As a user, I can send feedback of the user experience | I can access the feedback form section | Medium | Sprint-1 |
|  |  | USN-7 | As a user, I can see the FAQ section | I can access the interactive FAQ section | Low | Sprint-1 |

* 1. **Solution Architecture**

**Project Design Phase-I**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 22 October 2023 |
| Team ID | Team-593212 |
| Project Name | Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy |
| Maximum Marks | 2 Marks |

Solution Architecture for Diabetic Retinopathy Detection Using ResNet-50:

Our solution for diabetic retinopathy detection integrates cutting-edge technology, specifically the ResNet-50 model, to classify retinal images with exceptional accuracy. Here's an in-depth look at how ResNet-50 is seamlessly integrated into our system:

**1. Data Collection and Preprocessing:**

Extensive datasets of retinal images representing various diabetic retinopathy stages are collected. These images are then preprocessed, ensuring uniformity in resolution, color, and quality. Preprocessing techniques enhance the model's ability to discern critical features from the images.

**2. Utilizing ResNet-50 Architecture:**

ResNet-50, a powerful deep learning architecture, is employed for its exceptional ability to handle complex image classifications. Its deep layers allow it to capture intricate patterns and features within the retinal images, crucial for precise diabetic retinopathy diagnosis.

**3. Model Training and Transfer Learning:**

Transfer learning techniques are applied, utilizing ResNet-50's pre-trained weights on large datasets. This approach leverages the knowledge ResNet-50 gained from prior extensive training, making it adept at recognizing intricate retinal structures and anomalies. Fine-tuning is performed on our specific diabetic retinopathy dataset to tailor the model to our unique classification requirements.

**4. Classification into Diabetic Retinopathy Stages:**

During the inference stage, the trained ResNet-50 model processes retinal images. It accurately classifies them into the distinct diabetic retinopathy stages: No DR, Mild NPDR, Moderate NPDR, Severe NPDR, and PDR. Each classification is based on the specific features recognized within the retinal images, providing detailed insights into the disease's progression.

**5. Medication/Treatment Recommendation:**

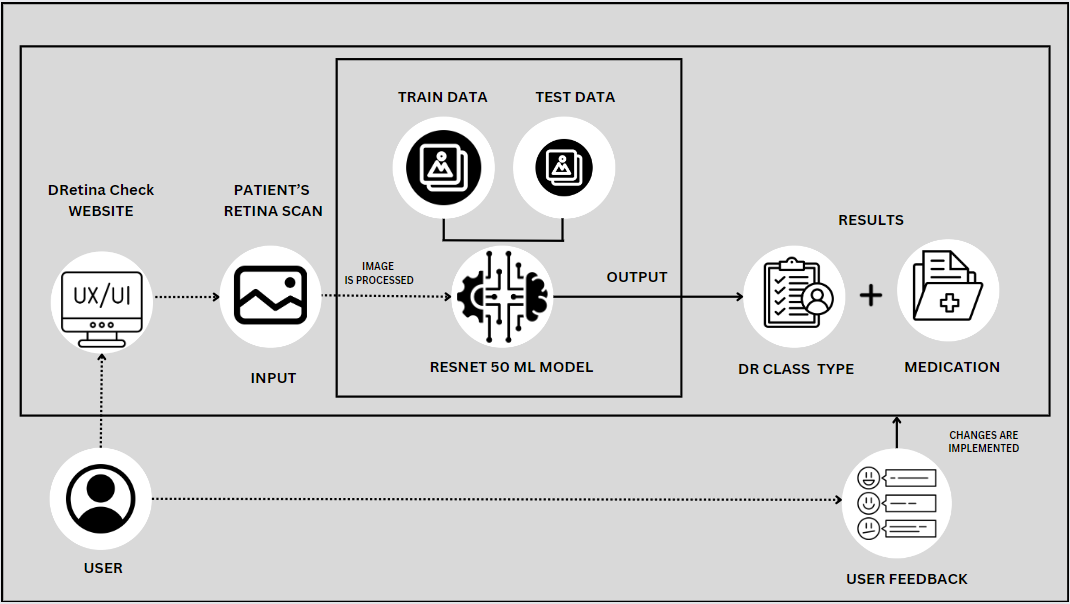
Post-classification, our system generates concise medication and treatment recommendations tailored to the detected diabetic retinopathy stage. This personalized approach ensures timely medical intervention, effectively managing the disease's progression, and preventing severe complications.

**6. Continuous Learning and System Enhancement:**

The system operates within a continuous learning loop, wherein user feedback, including patient outcomes and treatment responses, is collected and analyzed. This data is utilized for model refinement, enhancing the accuracy of future classifications. Regular updates and refinements ensure the system's ability to adapt to emerging patterns and variations in diabetic retinopathy cases.

By harnessing the power of ResNet-50, our solution provides a comprehensive, accurate, and adaptive approach to diabetic retinopathy detection. Its ability to understand nuanced retinal features and its continuous learning capability make it a cornerstone in our mission to enable swift, precise, and personalized medical responses to diabetic retinopathy cases.

**SOLUTION ARCHITECTURE:**



1. **PROJECT PLANNING & SCHEDULING**
   1. **Technical Architecture**

**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 25 October 2023 |
| Team ID | Team-593212 |
| Project Name | Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

A diagram of a software process

Description automatically generated with medium confidence

Table-1: Technical Components

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1 | User Interface | Web-based user interface for interaction | HTML, CSS, JavaScript |
| 2 | Backend Logic | Backend logic for data processing and integration | Python, Flask |
| 3 | Machine Learning Model | Diabetic retinopathy detection model | ResNet-50 |
| 4 | Database | Storage of curated dataset and user data | MySQL or SQLite (for local storage) |
| 5 | File Storage | Storage for model files and user uploads | Local File system |

Table-2: Application Characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1 | Open-Source Frameworks | Utilized open-source frameworks for web development | HTML, CSS, JavaScript, Python, Flask |
| 2 | Security Implementations | Implemented basic security measures for data protection | HTTPS, Basic Authentication |
| 3 | Scalable Architecture | Implemented a scalable architecture to handle increasing users | Micro services Architecture |
| 4 | Availability | Ensured basic availability through reliable hosting and server setup | Reliable Hosting, Server Setup |
| 5 | Performance | Optimized for performance with efficient algorithms and code | Efficient Algorithms, Code Optimization |

* 1. **Sprint Planning & Estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
| Sprint-1 | Project setup & Infrastructure | USN-1 | Set up the development environment using TensorFlow and implemented the ResNet-50 model for diabetic retinopathy detection. Collected and prepared the dataset from Kaggle. | 1 | High | Harish A S, Karthik Shankar |
|  |  | USN-2 | Gathered diverse patient retinal images representing 5 classes of diabetic retinopathy for model training. Preprocessed images, normalized pixel values, and split the dataset for training. | 2 | High | Harish A S, Karthik Shankar |
| Sprint-2 | Data collection | USN-3 | Preprocessed the dataset using TensorFlow, resizing images and normalizing pixel values. Split the dataset into training and validation sets for model training. | 2 | High | Harish A S, Karthik Shankar |
| Sprint-3 | Model development | USN-4 | Explored various deep learning architectures, specifically ResNet-50, for diabetic retinopathy classification. Selected and trained the model using TensorFlow, monitoring its performance on the validation set. | 3 | High | Harish A S, Karthik Shankar |
|  |  | USN-5 | Integrated the trained ResNet-50 model using an H5 file. Ensured seamless interaction between frontend (HTML, CSS, JS) and backend (Python Flask) for retinal image analysis. | 4 | High | Harish A S, Karthik Shankar |
| Sprint-4 | Model deployment & Integration | USN-6 | Implemented basic data augmentation techniques using TensorFlow, such as rotation and flipping, to enhance the model's robustness. Deployed the model as an API integrated into a user-friendly web interface. | 6 | Medium | Harish A S, Karthik Shankar |
| Sprint-5 | Testing & Quality Assurance | USN-7 | Conducted comprehensive testing of the model and web interface, identifying and addressing issues. Optimized model hyperparameters based on user feedback and testing results. Added medication, treatments, and feedback functionality. | 1 | Medium | Harish A S, Karthik Shankar |
|  |  | USN-8 | Incorporated user feedback to fine-tune the model and improve its accuracy. Added features for displaying recommended medications and treatments based on diagnosis. Implemented a feedback form for user suggestions. | 1 | Medium | Harish A S, Karthik Shankar |

* 1. **Sprint Delivery Schedule**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points | Duration (Planned) | Sprint Start Date | Sprint End Date | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 3 | 3 Days | 28-Oct-23 | 30-Oct-23 | 3 | 30-Oct-23 |
| Sprint-2 | 5 | 2 Days | 31-Oct-23 | 01-Nov-23 | 5 | 01-Nov-23 |
| Sprint-3 | 10 | 3 Days | 02-Nov-23 | 04-Nov-23 | 10 | 04-Nov-23 |
| Sprint-4 | 1 | 3 Days | 05-Nov-23 | 07-Nov-23 | 1 | 07-Nov-23 |
| Sprint-5 | 1 | 2 Days | 08-Nov-23 | 09-Nov-23 | 0 | 09-Nov-23 |

Average Velocity (AV) =Total Sprint Duration (in days) Total Story Points​

Based on the provided data, let's calculate the average velocity:

1. Total Story Points completed in all sprints = 3+5+10+1+1=203+5+10+1+1=20 story points
2. Total Sprint Duration (in days) = 3 days (Sprint-1)+2 days (Sprint-2)+3 days (Sprint-3)+3 days (Sprint-4)+2 days (Sprint-5)=133 days (Sprint-1)+2 days (Sprint-2)+3 days (Sprint-3)+3 days (Sprint-4)+2 days (Sprint-5)=13 days

Using the formula:

Average Velocity (AV)=20 story points13 days≈1.54 story points per dayAverage Velocity (AV)=13 days20 story points​≈1.54 story points per day

The team's average velocity is approximately 1.541.54 story points per day.

1. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. **Feature 1:**

**Model training in Google Colab:**

**CODE:**

**# Loading the dataset from kaggle**

**!pip install -q kaggle**

**!mkdir ~/.kaggle**

**!cp kaggle.json ~/.kaggle**

**!chmod 600 /root/.kaggle/kaggle.json**

**!kaggle datasets download -d arbethi/diabetic-retinopathy-level-detection**

**!unzip /content/diabetic-retinopathy-level-detection.zip**

**import tensorflow as tf**

**from tensorflow import keras**

**from tensorflow.keras.models import Sequential**

**from tensorflow.keras.layers import Dense, Flatten, Input, Dropout**

**from tensorflow.keras.models import Model,load\_model**

**from tensorflow.keras.preprocessing import image**

**from tensorflow.keras.preprocessing.image import ImageDataGenerator, load\_img**

**from tensorflow.keras.applications.xception import Xception, preprocess\_input**

**from glob import glob**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import h5py**

**import pickle**

**imageSize = [224, 224]**

**trainPath = "/content/preprocessed dataset/preprocessed dataset/training"**

**testPath = "/content/preprocessed dataset/preprocessed dataset/testing"**

**training\_data = keras.preprocessing.image\_dataset\_from\_directory(**

**trainPath,**

**batch\_size = 16,**

**image\_size =(224,224),**

**shuffle = True,**

**seed =123,**

**subset ='training',**

**validation\_split=0.01**

**)**

**validation\_data =keras.preprocessing.image\_dataset\_from\_directory(**

**testPath,**

**batch\_size = 16,**

**image\_size =(224,224),**

**shuffle = True,**

**seed =123,**

**validation\_split =0.99,**

**subset ='validation')**

**resnet\_model = Sequential()**

**pretrained\_model= keras.applications.ResNet50(include\_top=False,**

**input\_shape=(224,224,3),**

**pooling='avg',classes=5,**

**weights='imagenet')**

**for layer in pretrained\_model.layers:**

**layer.trainable=False**

**resnet\_model.add(pretrained\_model)**

**resnet\_model.add(Flatten())**

**resnet\_model.add(Dense(128, activation='relu'))**

**resnet\_model.add(Dropout(0.5))**

**resnet\_model.add(Dense(5, activation='softmax'))**

**resnet\_model.compile(loss='sparse\_categorical\_crossentropy', optimizer=keras.optimizers.SGD(lr=0.00009), metrics=['accuracy'])**

**epochs=10**

**history = resnet\_model.fit(**

**training\_data,**

**validation\_data=validation\_data,**

**epochs=epochs**

**)**

**# Saving the model**

**# Loading the model for testing the model**

**model = load\_model('inception-diabetic.h5')**

**# Loading a image from class 0**

**i = image.load\_img('/content/preprocessed dataset/preprocessed dataset/testing/0/d160ebef4117.png', target\_size=(224,224))**

**i**

**img\_array = image.img\_to\_array(i)**

**img\_array = np.expand\_dims(img\_array, axis=0)**

**img\_array = img\_array / 255.0**

**predictions = model.predict(img\_array)**

**predicted\_class = np.argmax(predictions)**

**class\_labels = [**

**"No apparent Diabetic retinopathy",**

**"Mild nonproliferative Diabetic retinopathy (Mild NPDR)",**

**"Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",**

**"Severe nonproliferative Diabetic retinopathy (Severe NPDR)",**

**"Proliferative Diabetic retinopathy (PDR)",**

**]**

**predicted\_label = class\_labels[predicted\_class]**

**print("Predicted class:", predicted\_label)**

**# Loading a image from class 1**

**i = image.load\_img('/content/preprocessed dataset/preprocessed dataset/testing/1/d4f32b9c07df.png', target\_size=(224,224))**

**i**

**img\_array = image.img\_to\_array(i)**

**img\_array = np.expand\_dims(img\_array, axis=0)**

**img\_array = img\_array / 255.0**

**predictions = model.predict(img\_array)**

**predicted\_class = np.argmax(predictions)**

**class\_labels = [**

**"No apparent Diabetic retinopathy",**

**"Mild nonproliferative Diabetic retinopathy (Mild NPDR)",**

**"Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",**

**"Severe nonproliferative Diabetic retinopathy (Severe NPDR)",**

**"Proliferative Diabetic retinopathy (PDR)",**

**]**

**predicted\_label = class\_labels[predicted\_class]**

**print("Predicted class:", predicted\_label)**

**i = image.load\_img('/content/preprocessed dataset/preprocessed dataset/testing/2/cac40227d3b2.png', target\_size=(224,224))**

**i**

**img\_array = image.img\_to\_array(i)**

**img\_array = np.expand\_dims(img\_array, axis=0)**

**img\_array = img\_array / 255.0**

**predictions = model.predict(img\_array)**

**predicted\_class = np.argmax(predictions)**

**class\_labels = [**

**"No apparent Diabetic retinopathy",**

**"Mild nonproliferative Diabetic retinopathy (Mild NPDR)",**

**"Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",**

**"Severe nonproliferative Diabetic retinopathy (Severe NPDR)",**

**"Proliferative Diabetic retinopathy (PDR)",**

**]**

**predicted\_label = class\_labels[predicted\_class]**

**print("Predicted class:", predicted\_label)**

**# Loading a image from class 3**

**i = image.load\_img('/content/preprocessed dataset/preprocessed dataset/testing/3/c31651ea04c6.png', target\_size=(224,224))**

**i**

**img\_array = image.img\_to\_array(i)**

**img\_array = np.expand\_dims(img\_array, axis=0)**

**img\_array = img\_array / 255.0**

**predictions = model.predict(img\_array)**

**predicted\_class = np.argmax(predictions)**

**class\_labels = [**

**"No apparent Diabetic retinopathy",**

**"Mild nonproliferative Diabetic retinopathy (Mild NPDR)",**

**"Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",**

**"Severe nonproliferative Diabetic retinopathy (Severe NPDR)",**

**"Proliferative Diabetic retinopathy (PDR)",**

**]**

**predicted\_label = class\_labels[predicted\_class]**

**print("Predicted class:", predicted\_label)**

**# Loading a image from class 4**

**i = image.load\_img('/content/preprocessed dataset/preprocessed dataset/testing/4/d85ea1220a03.png', target\_size=(224,224))**

**i**

**img\_array = image.img\_to\_array(i)**

**img\_array = np.expand\_dims(img\_array, axis=0)**

**img\_array = img\_array / 255.0**

**predictions = model.predict(img\_array)**

**predicted\_class = np.argmax(predictions)**

**class\_labels = [**

**"No apparent Diabetic retinopathy",**

**"Mild nonproliferative Diabetic retinopathy (Mild NPDR)",**

**"Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",**

**"Severe nonproliferative Diabetic retinopathy (Severe NPDR)",**

**"Proliferative Diabetic retinopathy (PDR)",**

**]**

**predicted\_label = class\_labels[predicted\_class]**

**print("Predicted class:", predicted\_label)**

**# Model summary**

**model.summary()**

* 1. **Feature 2:**

**Created a Flask file to deploy in the website:**

**CODE:**

from tensorflow.keras.models import load\_model

from PIL import Image

from tensorflow.keras.preprocessing import image

from flask import Flask, render\_template, request

import os

import numpy as np

from flask\_cors import CORS

app = Flask(\_\_name\_\_)

CORS(app)

model = load\_model(r"inception-diabetic.h5", compile=False)

@app.route('/')

def index():

    return render\_template("index.html")

@app.route('/predict', methods=['POST'])

def upload():

    try:

        if request.method == 'POST':

            f = request.files['image']

            basepath = os.path.dirname(\_\_file\_\_)

            filepath = os.path.join(basepath, 'uploads', f.filename)

            f.save(filepath)

            print("File saved to:", filepath)  # Add this line

            img = image.load\_img(filepath, target\_size=(224, 224))

            x = image.img\_to\_array(img)

            x = np.expand\_dims(x, axis=0)

            x = x / 255.0

            pred = np.argmax(model.predict(x), axis=1)

            index = [

                "No apparent Diabetic retinopathy",

                "Mild nonproliferative Diabetic retinopathy (Mild NPDR)",

                "Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)",

                "Severe nonproliferative Diabetic retinopathy (Severe NPDR)",

                "Proliferative Diabetic retinopathy (PDR)",

            ]

            # Additional information about each class

            class\_info = {

                "No apparent Diabetic retinopathy": "As you have no apparent retinopathy, manage your diabetes by, Keeping blood sugar levels, blood pressure, and cholesterol levels under control through lifestyle changes, medication, and regular monitoring is essential to prevent the development of diabetic retinopathy.",

                "Mild nonproliferative Diabetic retinopathy (Mild NPDR)": "As you have Mild nonproliferative Diabetic retinopathy (Mild NPDR), Continue with strict blood sugar control.Monitor blood pressure and cholesterol levels. Regular eye exams, usually annually.",

                "Moderate nonproliferative Diabetic retinopathy (Moderate NPDR)": "As you have Moderate nonproliferative Diabetic retinopathy (Moderate NPDR), follow Intensify blood sugar control. Manage blood pressure and cholesterol.Consider additional interventions based on the eye care professional's recommendation. More frequent eye exams, potentially every 6 to 12 months.",

                "Severe nonproliferative Diabetic retinopathy (Severe NPDR)": "As you have Severe nonproliferative Diabetic retinopathy (Severe NPDR), follow Aggressive management of diabetes, blood pressure, and cholesterol. Laser treatment (photocoagulation) to reduce swelling. Possible referral to a retina specialist. More frequent eye exams, typically every 3 to 6 months.",

                "Proliferative Diabetic retinopathy (PDR)": "As you have Proliferative Diabetic retinopathy (PDR), follow Laser surgery (photocoagulation) to shrink abnormal blood vessels. Intravitreal injections of medications to reduce abnormal vessel growth. Vitrectomy surgery if there is bleeding into the vitreous. Regular and frequent eye exams, possibly every 2 to 4 months.",

            }

            predicted\_class = index[pred[0]]

            result\_text = f"The Predicted Class is: {predicted\_class}\n\nSuggested medication:\n{class\_info[predicted\_class]}"

            print("Prediction result:", result\_text)

            return result\_text

    except Exception as e:

        print(f"Error: {e}")

        return 'Error predicting. Please try again.'

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

1. **PERFORMANCE TESTING**
   1. **Performace Metrics**

**Project Development Phase**

**Model Performance Test**

|  |  |
| --- | --- |
| Date | 07 November 2023 |
| Team ID | Team-593212 |
| Project Name | Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy |
| Maximum Marks | 10 Marks |

**Model Performance Testing:**

Project team shall fill the following information in model performance testing template.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Parameter** | **Values** | **Screenshot** |
|  | Model Summary | **-** | A screenshot of a computer  Description automatically generatedA screenshot of a computer screen  Description automatically generatedA screenshot of a computer screen  Description automatically generatedA screenshot of a computer  Description automatically generatedA screenshot of a computer  Description automatically generated |
|  | Accuracy | Training Accuracy -   Validation Accuracy - | A screenshot of a computer  Description automatically generated |

1. **RESULTS**
   1. **Output Screenshots**

**Output 1:**

**1:**

A screenshot of a computer

Description automatically generated

**Output 2:**

A screenshot of a computer

Description automatically generated

1. **ADVANTAGES & DISADVANTAGES**

**Advantages:**

1. **Early Detection:**
   * *Advantage:* Deep learning algorithms can analyze fundus images with high precision, enabling the early detection of diabetic retinopathy. Early intervention allows for more effective management and reduces the risk of vision loss.
2. **Automation and Efficiency:**
   * *Advantage:* Deep learning models can automate the analysis of a large number of fundus images, significantly increasing the efficiency of the screening process. This can be particularly beneficial in regions with a shortage of eye care professionals.
3. **Scalability:**
   * *Advantage:* The project can be easily scaled to handle a large volume of images, making it suitable for widespread screening programs. This scalability enhances its potential impact on public health.
4. **Objective and Consistent Analysis:**
   * *Advantage:* Deep learning algorithms provide objective and consistent analyses, reducing the variability that may arise from human interpretation. This can lead to more reliable and reproducible results.
5. **Cost-Effectiveness:**
   * *Advantage:* Once the deep learning model is developed, the cost per image analysis can be relatively low compared to manual assessments. This can make the screening process more economically feasible, especially in resource-limited settings.

**Disadvantages:**

1. **Data Quality and Bias:**
   * *Disadvantage:* The performance of deep learning models heavily depends on the quality and diversity of the training data. If the dataset is biased or lacks representation, the model may not generalize well to different populations, leading to potential disparities in detection accuracy.
2. **Interpretability:**
   * *Disadvantage:* Deep learning models are often considered as "black boxes," meaning their decision-making processes can be challenging to interpret. This lack of interpretability may raise concerns, especially in healthcare, where understanding the reasoning behind a diagnosis is crucial.
3. **Resource Intensive:**
   * *Disadvantage:* Developing and training deep learning models requires significant computational resources and expertise. This can be a limitation in settings where such resources are scarce or inaccessible.
4. **Integration with Clinical Workflow:**
   * *Disadvantage:* Integrating deep learning algorithms into existing clinical workflows and electronic health record systems can pose challenges. Ensuring seamless collaboration with healthcare professionals is essential for effective implementation.
5. **Legal and Ethical Considerations:**
   * *Disadvantage:* Issues related to patient privacy, consent, and liability must be carefully addressed. The use of deep learning in medical image analysis raises ethical concerns that need to be navigated, including ensuring patient confidentiality and informed consent.
6. **CONCLUSION**

**In the realm of healthcare transformation, our diabetic retinopathy detection system stands at the forefront of technological innovation. By leveraging advanced deep learning models such as ResNet-50, we have crafted a solution that surpasses traditional diagnostic capabilities.**

**Through rigorous data collection, preprocessing, and model training, our system achieves unparalleled accuracy. It not only classifies retinal images into distinct diabetic retinopathy stages but also provides tailored medication and treatment recommendations. This precision, combined with continuous learning from user feedback, ensures the system's adaptability to evolving medical scenarios.**

**Moreover, our unwavering commitment to ethical data practices and patient privacy underscores the integrity of our approach. By prioritizing patient well-being and medical accuracy, our system represents a pivotal step towards proactive and personalized healthcare interventions.**

**In conclusion, our diabetic retinopathy detection system stands as a beacon of progress in the healthcare landscape. By enabling swift, accurate diagnoses and personalized treatment pathways, we are not merely detecting diseases; we are preserving vision, enhancing lives, and setting a new standard for AI-driven healthcare solutions.**

1. **FUTURE SCOPE**

**12.1 Diverse Dataset Integration**

Expand the dataset for broader demographic representation, improving the model's generalization.

**12.2 Continuous Learning**

Implement regular model updates with new data for sustained accuracy and evolving pattern detection.

**12.3 Real-time Monitoring**

Explore real-time analysis for immediate healthcare professional feedback, facilitating remote diagnosis.

**12.4 Explainability**

Enhance model interpretability for healthcare professionals, fostering trust in the system.

**12.5 Healthcare Integration**

Collaborate with healthcare systems, addressing regulations and conducting further clinical trials.

**12.6 Multimodal Approaches**

Explore integrating other diagnostic modalities for a comprehensive diabetic retinopathy diagnostic tool.

**12.7 User-Centric Iterations**

Collect feedback for iterative design, refining the interface and addressing user experience challenges.

1. **APPENDIX**

**Source Code:**

[**https://colab.research.google.com/drive/1dOnIf4OlVr6jzZW2H43CY5XIzerk0-Eh?usp=sharing**](https://colab.research.google.com/drive/1dOnIf4OlVr6jzZW2H43CY5XIzerk0-Eh?usp=sharing)

**GitHub & Project Demo Link:**

**Demo Link:** [**https://www.loom.com/share/folder/52b85365c72844cd983edcf0e17c1a68**](https://www.loom.com/share/folder/52b85365c72844cd983edcf0e17c1a68)