Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Prepared By

V.Anitha

Assistant Professor

Department of Information Technology

National Engineering College

Kovilpatti

ABSTRACT

Diabetic Retinopathy (DR) presents a significant threat to the vision health of individuals with diabetes mellitus, underscoring the crucial necessity for early detection to mitigate potential irreversible consequences. This study explores the incorporation of deep learning techniques into the analysis of fundus images to create a robust and effective system for detecting DR at an early stage. The proposed work offers a standardized and objective approach to DR detection, reducing reliance on individual expertise and facilitating consistent, accurate results. The deep learning model presented here, learns intricate patterns and features of early DR features, allowing it to generalize effectively to unseen data. Rigorous validation and testing procedures are implemented to ensure the model's reliability and generalizability across diverse patient demographics and imaging conditions. Through the fusion of medical knowledge and technological innovation, this initiative seeks to play a role in the progression of ophthalmic care, highlighting the revolutionary possibilities of utilizing deep learning for early detection of Diabetic Retinopathy (DR). The results of the research not only offer the potential to improve diagnostic accuracy but also underscore the crucial influence of artificial intelligence in molding the trajectory of healthcare, particularly in addressing the challenges associated with diabetic retinopathy.

CHAPTER 1

INTRODUCTION

Diabetic Retinopathy (DR) is a debilitating eye condition that arises as a complication of diabetes and remains a significant global health concern. It is a progressive disease that affects the blood vessels of the retina, leading to vision impairment and blindness if left untreated. Early detection and timely intervention are crucial in managing and preventing the progression of diabetic retinopathy. With advancements in medical imaging technology, particularly in ophthalmology, the field of Deep Learning (DL) has emerged as a powerful tool for the automated analysis of medical images, including fundus images of the retina. Fundus images provide a detailed view of the back of the eye, allowing for the assessment of retinal health and the early identification of diabetic retinopathy.

This project focuses on leveraging Deep Learning techniques for the early detection of diabetic retinopathy through the analysis of fundus images. The goal is to develop an intelligent system capable of automatically classifying retinal images into different stages of diabetic retinopathy, ranging from mild to severe. By doing so, the project aims to contribute to the enhancement of early diagnosis and intervention strategies, ultimately improving patient outcomes.

Diabetic Retinopathy

Diabetic retinopathy (DR) is a serious and progressive eye condition that affects individuals with diabetes. It occurs when high levels of blood sugar damage the blood vessels in the retina, the light-sensitive tissue at the back of the eye. As DR advances, it can lead to vision impairment and, if left untreated, even blindness. Here are key aspects of diabetic retinopathy:

1. **Prevalence and Risk Factors:**
    - DR is a common complication of diabetes, affecting a significant number of individuals with both type 1 and type 2 diabetes.
    - Prolonged periods of uncontrolled blood sugar levels increase the risk of developing diabetic retinopathy.
2. **Stages of Diabetic Retinopathy:**
    - **Non-proliferative diabetic retinopathy (NPDR):** The early stage characterized by microaneurysms, small hemorrhages, and fluid leakage.

- **Proliferative diabetic retinopathy (PDR):** The advanced stage involving the growth of abnormal blood vessels on the retina, which can lead to severe vision loss.

3. **Symptoms:**

- In the early stages, diabetic retinopathy may not present noticeable symptoms.
- As the condition progresses, individuals may experience blurred vision, floaters, difficulty seeing at night, and, in advanced stages, sudden vision loss.

4. **Diagnostic Methods:**

- Regular eye exams, especially for individuals with diabetes, are crucial for early detection.
- Fundus photography and optical coherence tomography (OCT) are common diagnostic tools used to assess the retina's health.

5. **Importance of Early Detection:**

Early detection is key to preventing vision loss. Routine eye screenings can identify diabetic retinopathy in its early stages when interventions are more effective.

6. **Technological Advancements:**

The integration of artificial intelligence, particularly deep learning algorithms, has shown promise in automating the analysis of retinal images, aiding in early detection.

7. **Public Health Implications:**

Diabetic retinopathy is a significant public health concern globally, and its prevalence is expected to rise with the increasing incidence of diabetes.

8. **Preventive Measures:**

Managing diabetes through lifestyle modifications, regular medical check-ups, and adherence to prescribed medications is essential for preventing diabetic retinopathy.

Role of Deep Learning in Medical Imaging

The potential of Deep Learning (DL) in the realm of medical imaging holds immense promise, revolutionizing the way we approach diagnostics and patient care. DL, a subset of artificial intelligence, has demonstrated remarkable capabilities in extracting complex patterns and features from medical images, leading to more accurate and efficient diagnoses. This paradigm shift in medical imaging is driven by several key factors.

Firstly, DL algorithms can autonomously learn and adapt to intricate patterns within medical images, eliminating the need for explicit feature engineering. This adaptability allows the models to continually improve their performance with more extensive datasets and experience, enhancing their diagnostic accuracy over time.

Secondly, the deep neural networks employed in DL can handle vast amounts of heterogeneous data, such as diverse imaging modalities and patient demographics. This versatility makes DL particularly robust in handling the complexity inherent in medical datasets, allowing for a more comprehensive and personalized approach to patient care.

The integration of DL into medical imaging workflows has the potential to significantly reduce the time and effort required for image analysis. Automated image interpretation not only accelerates the diagnostic process but also alleviates the burden on healthcare professionals, enabling them to focus on more complex decision-making tasks.

Furthermore, DL models exhibit a capacity for hierarchical feature representation, enabling them to discern subtle abnormalities or early signs of diseases that may elude traditional image analysis methods.

Objective of the Project

The main goal of this project is to create a Deep Learning system for analyzing fundus images that can autonomously detect the initial indicators of Diabetic Retinopathy (DR). By utilizing a diverse dataset covering different DR stages, the deep learning model is designed to grasp subtle patterns characteristic of DR, providing a standardized and unbiased method for diagnosis. This study aims not only to push technological boundaries but also to tackle significant challenges like inconsistencies among observers and the resource-intensive nature of manual evaluations.

Significance of the Project:

Early Intervention: Early detection of diabetic retinopathy is crucial for timely intervention and preventing vision loss in diabetic patients.

Resource Optimization: Automated analysis through deep learning can assist healthcare professionals in efficiently managing a large volume of fundus images, optimizing resource allocation, and reducing diagnostic time.

Global Impact: The project has the potential to make a significant impact on a global scale by providing a cost-effective and scalable solution for diabetic retinopathy screening, especially in regions with limited access to specialized healthcare services.

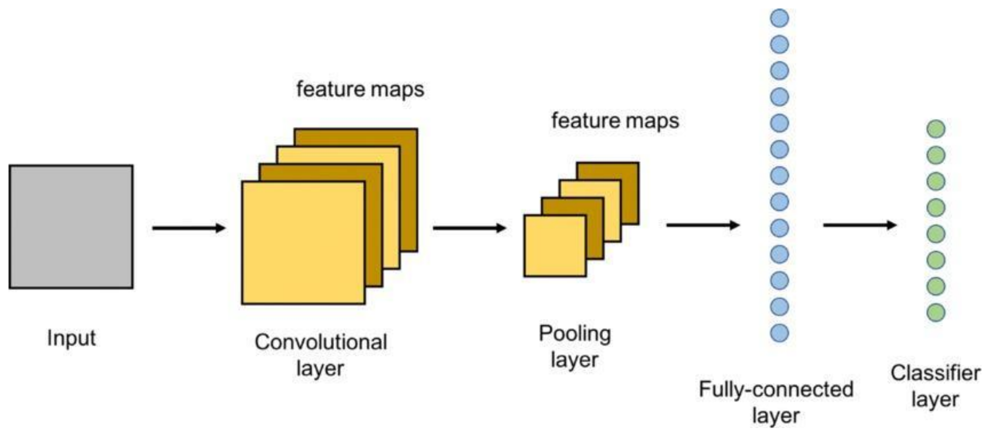
CHAPTER 2

LITERATURE REVIEW

This section explores different approaches and techniques employed in the image analysis for the early identification of diabetic retinopathy.

CNN

CNN stands for Convolutional Neural Network. It is a specialized type of artificial neural network designed for processing and analyzing visual data, such as images and videos. Some popular CNN architectures include AlexNet, VGG, ResNet, InceptionNet (GoogLeNet), and DenseNet, each with its own unique architectural characteristics and contributions to the field of computer vision. Upsampling involves interpolating new data points between existing data points to expand the dataset. In the context of image processing, upsampling is used to increase the dimensions of an image. This is typically done to match the resolution of another image, to prepare the image for further processing, or to display it on a higher-resolution screen. Upsampling is also used in signal processing to increase the sample rate of a signal, which can be beneficial in certain applications like audio processing or digital communication. In the context of machine learning, upsampling is used in certain data augmentation techniques, particularly in dealing with imbalanced datasets.



U-Net :

U-Net is a widely used convolutional neural network (CNN) architecture developed for semantic segmentation tasks .The architecture is designed to classify each pixel of an image into different classes or categories. The UNet architecture consists of two main parts: the contracting path and the expansive path. The contracting path is similar to a traditional convolutional neural network (CNN) with convolutional layers and pooling operations. It is responsible for capturing the context and features from the input image.The expansive path, on the other hand, is used to enable precise localization by upsampling and combining the feature maps obtained from the contracting path.The upsampling is typically done using transpose convolutions or other upsampling techniques, which help increase the resolution of the feature maps.The unique aspect of UNet is the skip connections that connect corresponding layers in the contracting and expansive paths. These skip connections allow information from earlier layers to be directly fed into later layers, which helps in preserving fine-grained details during the upsampling process. This characteristic of UNet enables it to create more accurate segmentation masks, as it fuses low-level and high-level features effectively.

SegNet:

SegNet is a deep convolutional neural network designed for semantic image segmentation. SegNet utilizes an encoder-decoder structure with skip connections similar to U-Net. Unlike U-Net, SegNet uses a smaller number of learnable parameters, making it computationally efficient. It is widely used for real-time segmentation tasks, such as self-driving cars and robotics, where efficiency is crucial.It consists of an encoder network that gradually reduces the spatial resolution and an asymmetric decoder network that upsamples the features. The encoder uses convolutional layers with max-pooling to extract hierarchical features, while the decoder uses upsampling and convolutional layers to restore the spatial resolution. SegNet has skip connections between corresponding layers in the encoder and decoder to retain spatial information during upsampling.

AlexNet:

AlexNet is a pioneering deep convolutional neural network. It was the winning model in the ILSVRC 2012 competition, revolutionizing computer vision. AlexNet's architecture includes multiple convolutional layers, max-pooling layers, and fully connected layers. It also introduced the concept of using ReLU activation functions and dropout regularization to enhance training and prevent overfitting. AlexNet was a breakthrough that paved the way for the subsequent development of more complex and powerful deep learning architectures.It consists of 8 layers, including 5 convolutional layers and 3 fully connected layers. AlexNet introduced novel concepts such as ReLU activation, dropout regularization, and data augmentation, which contributed to its superior performance at the time and paved the way for modern deep learning architectures.

InceptionNet :

InceptionNet, also known as GoogleNet, is a deep convolutional neural network architecture. The primary feature of InceptionNet is the use of inception modules, which consist of multiple parallel convolutional filters of different sizes. These modules allow the network to capture features at different scales efficiently. InceptionNet achieved high accuracy in image classification tasks with reduced computational complexity, making it popular in various applications. The inception module uses multiple convolutional filter sizes (1x1, 3x3, 5x5) and pooling operations in parallel, capturing features at different scales. This design reduces the number of parameters and computational complexity compared to traditional architectures while maintaining high accuracy. InceptionNet played a significant role in the development of modern convolutional neural networks.

CHAPTER 3

DETECTION OF DIABETIC RETINOPATHY

This chapter focuses on design of proposed methodology for the detection of diabetic retinopathy through the application of deep learning.

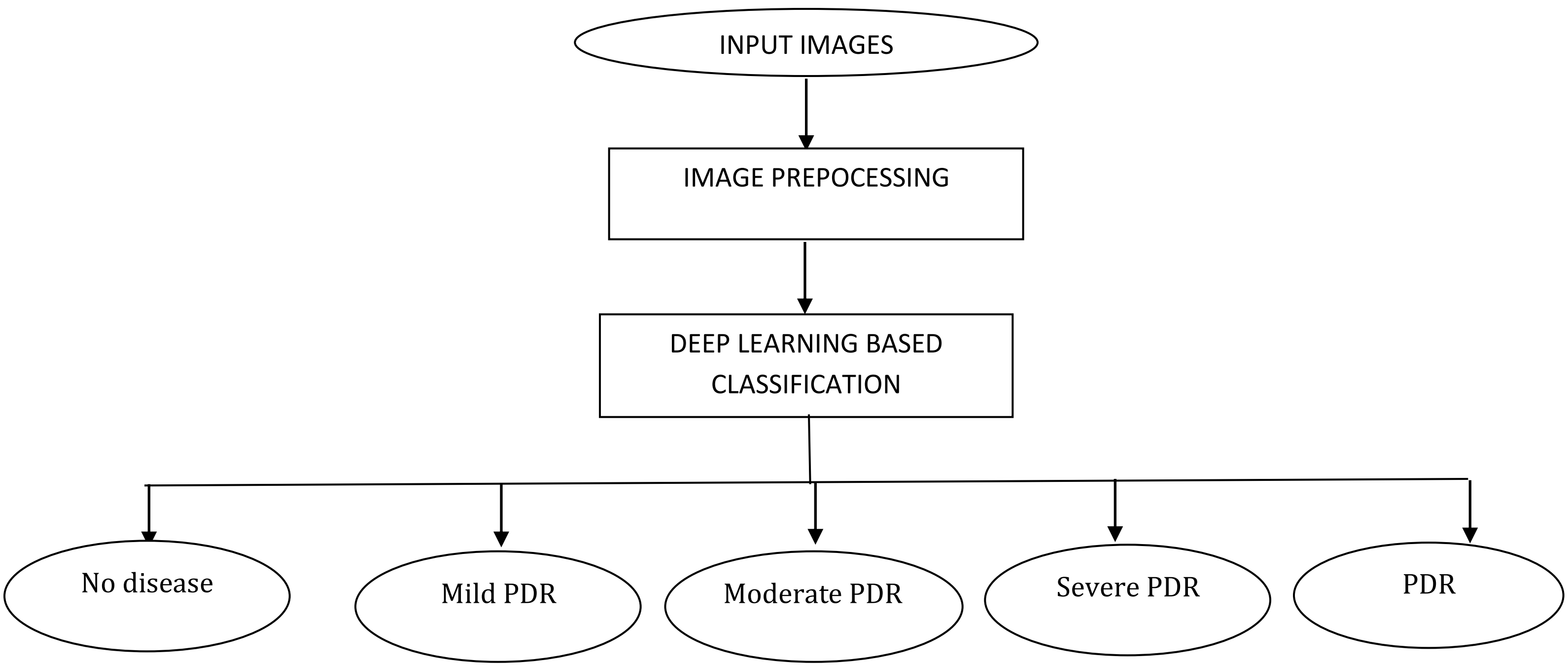


Fig 3.1 Architecture of Diabetic Retinopathy Classification

Initially data obtained from the dataset underwent processing to create training and testing sets. An Xception Net model was constructed using SoftMax as the activation function. Subsequently, predictions were made using the testing images within a Python Flask Application. Model performance was evaluated by comparing accuracy and loss metrics.

The collection of data holds paramount importance in constructing deep learning models, especially in endeavors such as the detection of diabetic retinopathy. For this project, data collection encompasses the compilation of a dataset comprising retinal fundus images, integral for the training and evaluation of the deep learning model.

Following data collection, Image preprocessing becomes a pivotal stage in readying raw retinal fundus images for training deep learning models. Ensuring uniformity, images are resized to a consistent resolution, aiding streamlined processing during model training. Normalizing pixel values to a standard scale, usually between 0 and 1, is crucial. Additionally, cropping images to emphasize pertinent regions, like the optic disc and macula, is undertaken. This not only diminishes computational complexity but also directs the model's attention to pivotal areas for diabetic retinopathy diagnosis.

The dataset is then divided into two parts training and testing dataset. The model is trained using the training set, while the testing set assesses the model's efficacy on data it has not been exposed to previously. Subsequently, the construction of the Xception model takes place. This deep convolutional neural network (CNN) architecture has showcased robust performance in tasks related to images. Distinguished by its considerable depth and the incorporation of depthwise separable convolutions, the Xception model excels in capturing intricate hierarchical features.

The layers of the pre-trained Xception model undergo freezing, achieved by iterating through them and setting layer.trainable = False. This freezing action serves to prevent the weights of these layers from being adjusted during the initial training phase, thereby preserving the knowledge acquired from ImageNet. Following the Xception base, a Global Average Pooling 2D layer is introduced. This layer contributes to the reduction of spatial dimensions in the feature maps by calculating the average value for each feature map.

Finally, the raw output scores (logits) of the network undergo transformation through the softmax activation function. This function normalizes the logits, ensuring that the probabilities collectively add up to 1 across all classes. This facilitates the model in making probabilistic predictions, ultimately selecting the class with the highest probability as the final prediction.

Accuracy is a measure of the overall correctness of the model's predictions. It is calculated as the ratio of correctly predicted instances to the total number of instances.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \qquad (1)$$

Loss is a measure of the model's prediction error. It quantifies how well the predicted values match the true values. The goal during training is to minimize the loss.

CHAPTER 4

IMPLEMENTATION

This chapter focuses on the software requirements and python libraries needed to implement the proposed methodology.

The proposed methodology is implemented based on the following flow,

- Downloading the dataset
- Model Building
- Cloudant DB
- Application Building

Dataset is downloaded from Kaggle, and the images are preprocessed after which testing and training images are separated. Model is built using Xception and Softmax. CloudantDB is created and a Python Flask application is used to build the web application.

As the training and testing folders are presented in the project dataset folder avariable is assigned and the folder path is invoked using it. Among all the four different transfer learning models (Xception) model is selected. The image input size of xception model is 299 x 299.

```
imageSize = [299, 299]

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

testPath = r"/content/preprocessed dataset/preprocessed dataset/testing"
```

The necessary libraries are imported as shown in the below figure.

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import 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
```

ImageDataGenerator class is instantiated and the configuration for the types of data augmentation. An instance of the ImageDataGenerator class can be constructed for training and testing.

```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

To make sure the weights does not get updated after each epoch we train our own model. considering the images of dimensions as 229 x 229 x 3. The parameter include_top ios assigned as False as convolution layer is used for feature extraction. The Flatten layer flattens the input. It does not affects the batch size.

```
xception = Xception(input_shape=imageSize + [3], weights='imagenet',include_top=False)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xcep
_kernels_notop.h5
83689472/83683744 [==============================] - 1s 0us/step
83697664/83683744 [==============================] - 1s 0us/step

# don't train existing weights
for layer in xception.layers:
    layer.trainable = False

# our layers - you can add more if you want
x = Flatten()(xception.output)
```

A dense layer is a deeply connected neural network layer. It is the most common and frequently used layer. An object named model is created with inputs as “xception”, input and output as dense layers**.**

```
prediction = Dense(5, activation='softmax')(x)

# create a model object
model = Model(inputs=xception.input, outputs=prediction)
```

The Dense layer is configured with a neuron count equal to the number of classes within the training set. In the final Dense layer, the neurons utilize softmax activation to transform their outputs into corresponding probabilities.

The compilation marks the conclusive stage in model creation, paving the way for the subsequent training phase. The loss function plays a pivotal role in identifying errors or deviations during the learning process. Optimization is a crucial step that refines the input weights by aligning predictions with the loss function for which the ‘adam’ optimizer is used in prediction.

```
# tell the model what cost and optimization method to use
model.compile(
  loss='categorical_crossentropy',
  optimizer='adam',
  metrics=['accuracy']
)
```

The model undergoes training for 30 epochs, and after each epoch, the current state of the model is saved if it exhibits the lowest encountered loss thus far. The training loss demonstrates a consistent decrease in nearly every epoch until the 10th epoch, suggesting potential for further model improvement.

fit_generator functions used to train a deep learning neural network

The model is saved under .h5 extension, representing a file format in the Hierarchical Data Format (HDF). This format encompasses multidimensional arrays storing scientific data.

```
model.save('Updated-Xception-diabetic-retinopathy.h5')
```

The user first logins into the IBM Cloud account and then navigate to the Catalog. To create the connection information needed by the application to connect to the instance, "New credential" is clicked. A name for the new credential is entered in the Add new credential window. The Manager role is accepted. A service ID is created or is automatically generated.Inline configuration parameters are added. It should be noted that this parameter isn't used by IBM Cloudant service credentials, so it is ignored. "Add" is clicked. The chevron is clicked to see the credentials required to access the service. To manage a connection from a local system, the connection is first initialized by constructing a Cloudant client. The cloudant library needs to be imported for this. IBM Cloud Identity & Access Management enables users to be securely authenticated and access control to be consistently controlled for all cloud resources in the IBM Bluemix Cloud Platform.

In python flask HTML files, CSS and required images are placed in template and static folder. All the necessary libraries are imported as shown in the figure below,

```python
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
import requests
from flask import Flask, request, render_template, redirect, url_for
from cloudant.client import Cloudant
```

The saved model is then loaded. The flask module must be then imported. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (__name__) as argument.

```python
model = load_model(r"Updated-Xception-diabetic-retinopathy.h5")

app = Flask(__name__)
```

A database is created using an initiated client.

```python
from cloudant.client import Cloudant

# Authenticate using an IAM API key
client = Cloudant.iam('username','apikey', connect=True)

# Create a database using an initialized client
my_database = client.create_database('my_database')
```

Based on the information provided by the user in the registration form, we store it in a data dictionary. Subsequently, data validation is performed using the _id parameter and the user input, which is stored in a query variable. The validation process involves passing the query variable into the my_database.get_user_result() method. The length of the resulting docs is checked using the len(docs.all()) function. If the docs length is 0, the user is successfully registered on the platform, and their data is stored in the database. Otherwise, a message is displayed, indicating that the user is already registered, and they are prompted to log in to use our web application for diabetic retinopathy prediction.

Similarly, in the login process, the user id and password entered in the login form are stored in the (user, pwd) variables. Credential validation is then carried out using the _id parameter and the user input, stored in a query variable. The validation involves passing the query variable into the my_database.get_user_result() method. The length of the resulting docs is checked using len(docs.all()). If the length of doc is 0, it indicates that the username is not found. Otherwise, the stored data in the database is validated, checking the username and password. If they match, the user can successfully log in and use our web application for diabetic retinopathy prediction. Otherwise, the user is prompted to provide correct credentials.

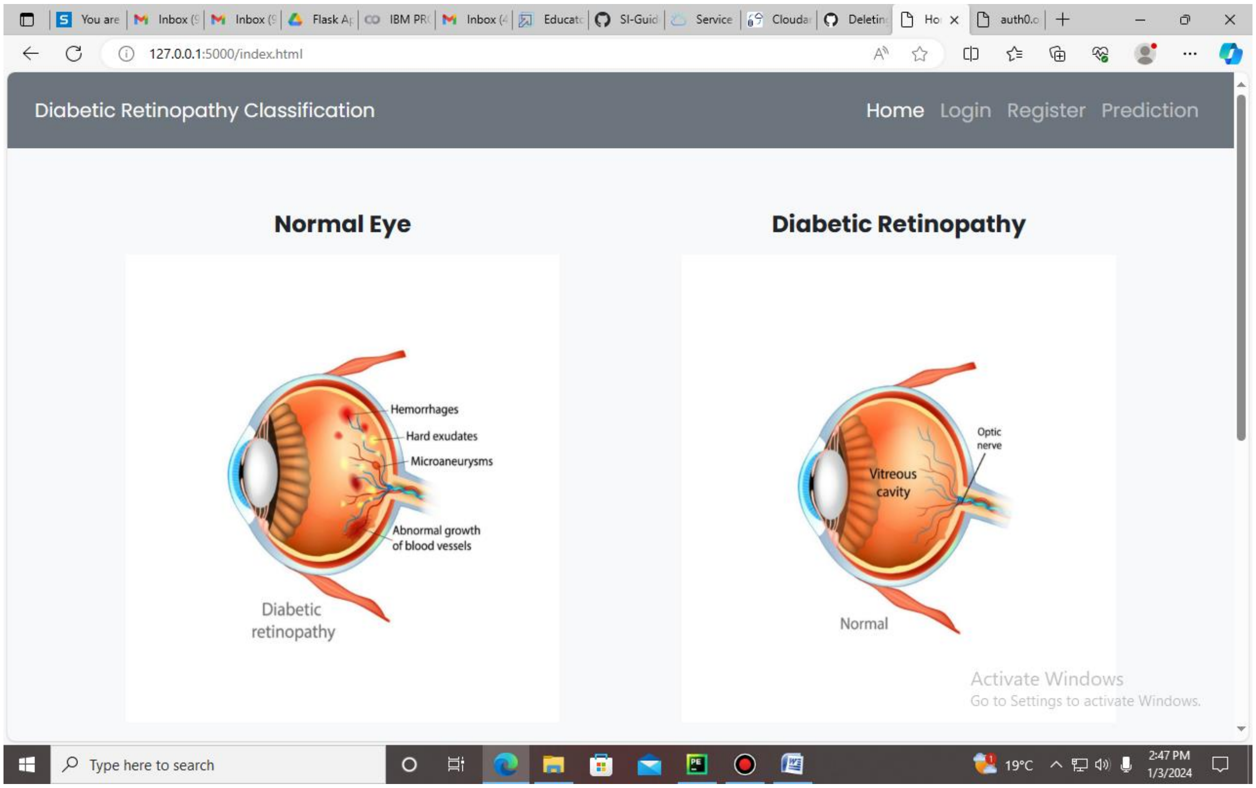
```python
#login page
@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/afterlogin',methods=['POST'])
def afterlogin():
    user = request.form['_id']
    passw = request.form['psw']
    print(user,passw)

    query = {'_id': {'$eq': user}}

    docs = my_database.get_query_result(query)
    print(docs)

    print(len(docs.all()))


    if(len(docs.all())==0):
        return render_template('login.html', pred="The username is not found.")
    else:
        if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
            return redirect(url_for('prediction'))
        else:
            print('Invalid User')
```

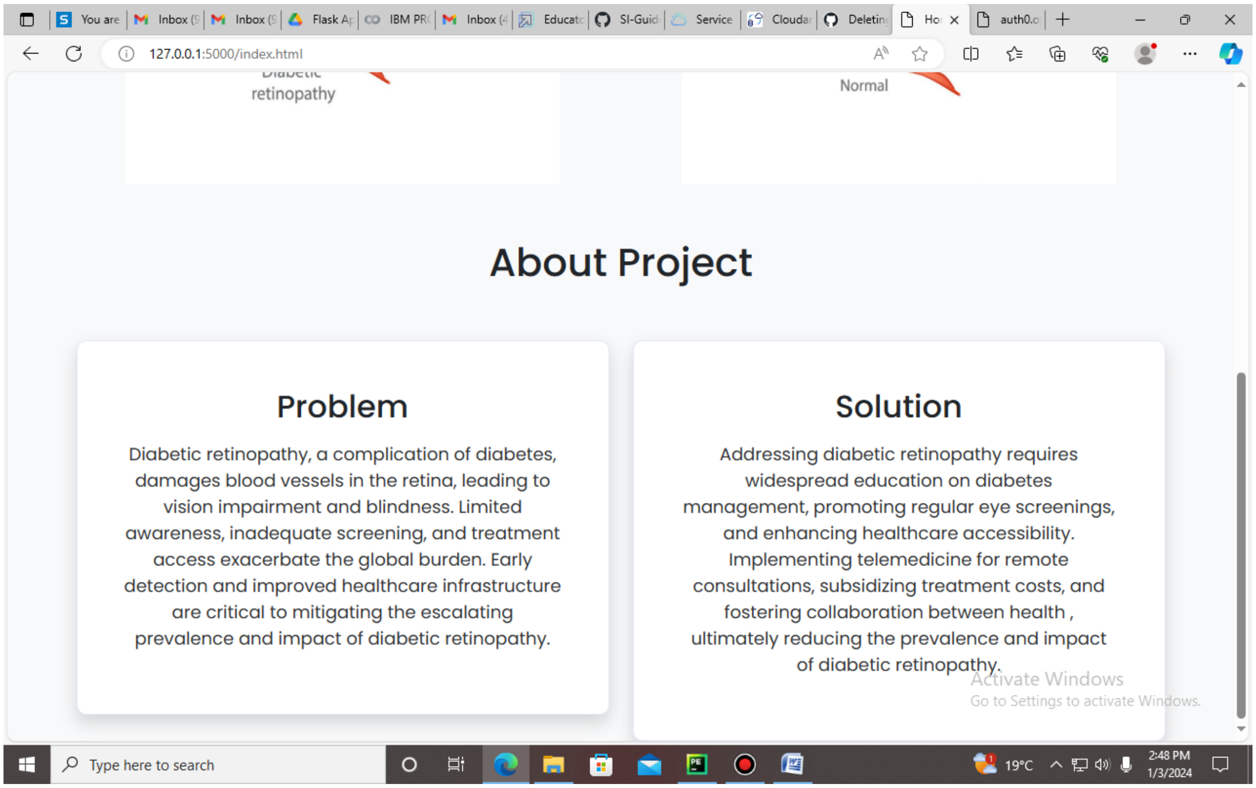
An image is chosen from the uploads folder, and it is loaded and resized using the load_img() method. Conversion of the image into an array is achieved through the use of the img_to_array() method, and its dimensions are expanded using the expand_dims() method. The processed input is then fed into the Xception model, and predictions for class probabilities are made using the predict() method. The class with the highest probability is determined by employing np.argmax.

CHAPTER 5

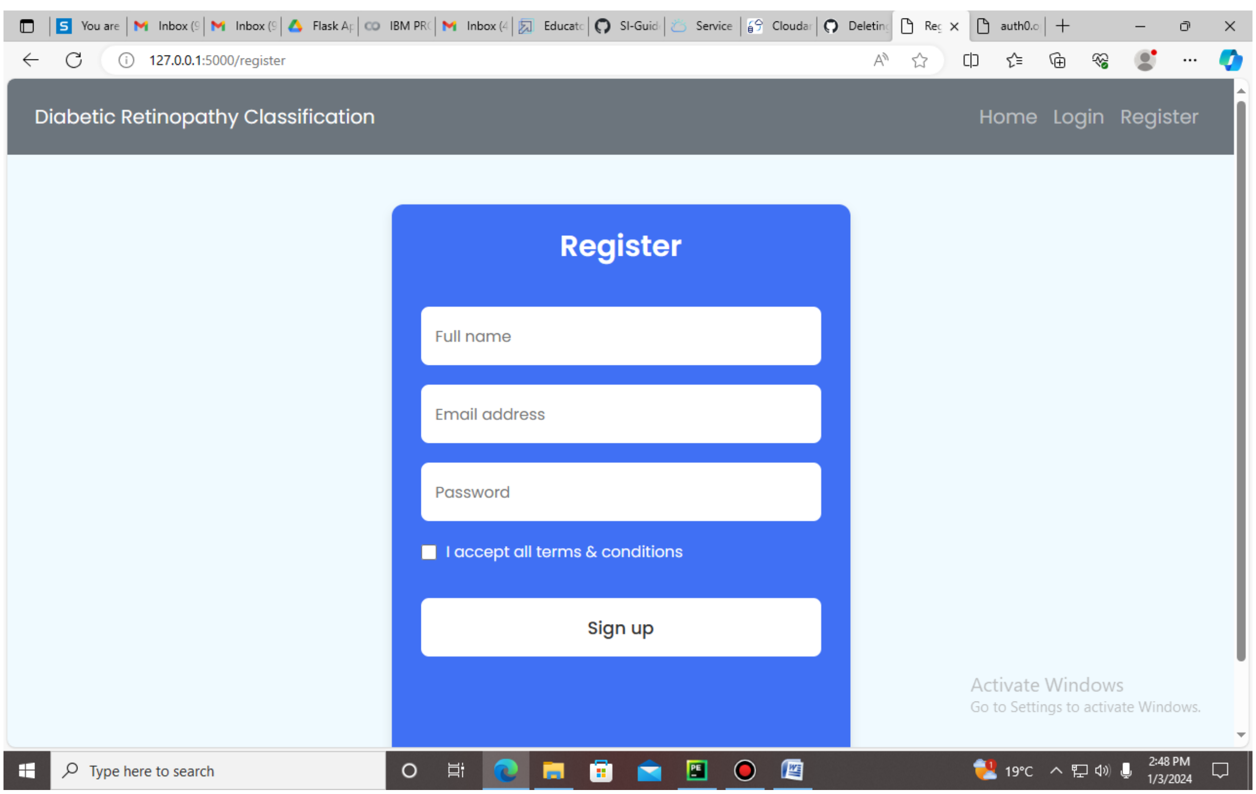
RESULTS AND DISCUSSION

<u>Index Page</u>

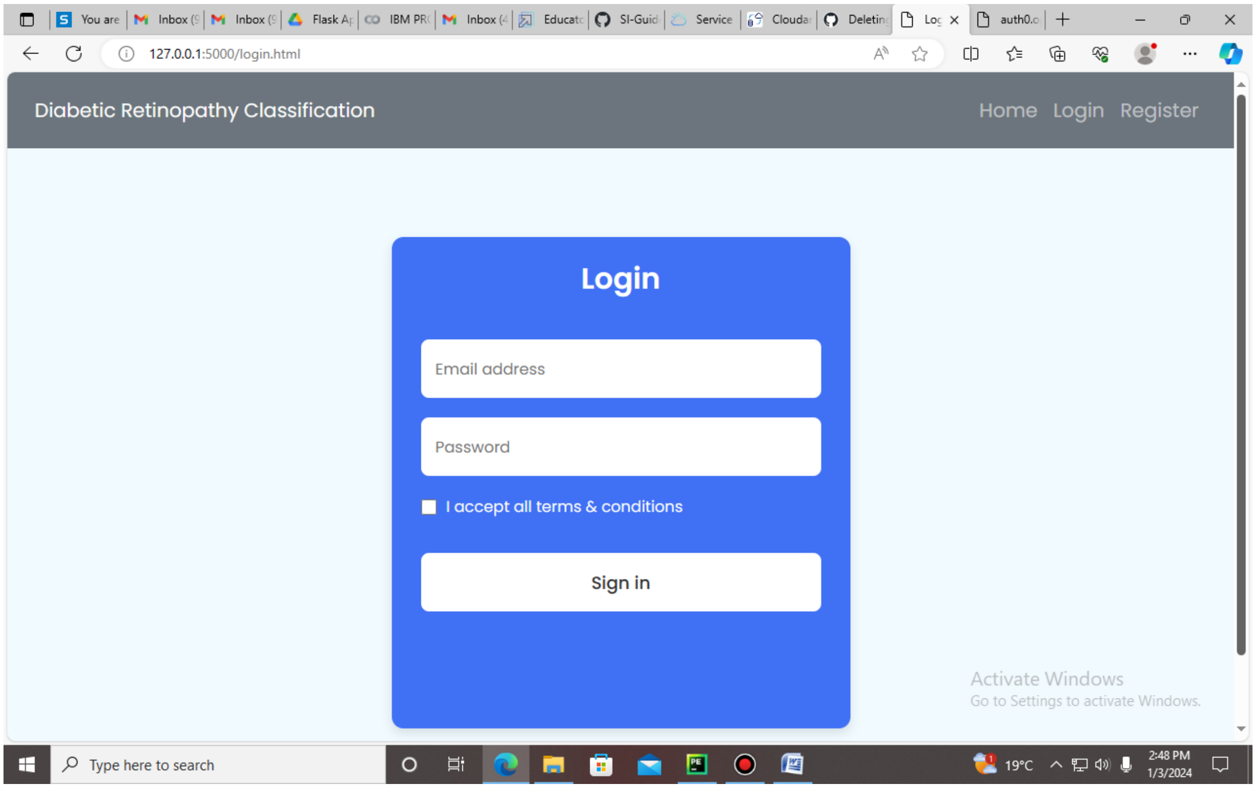




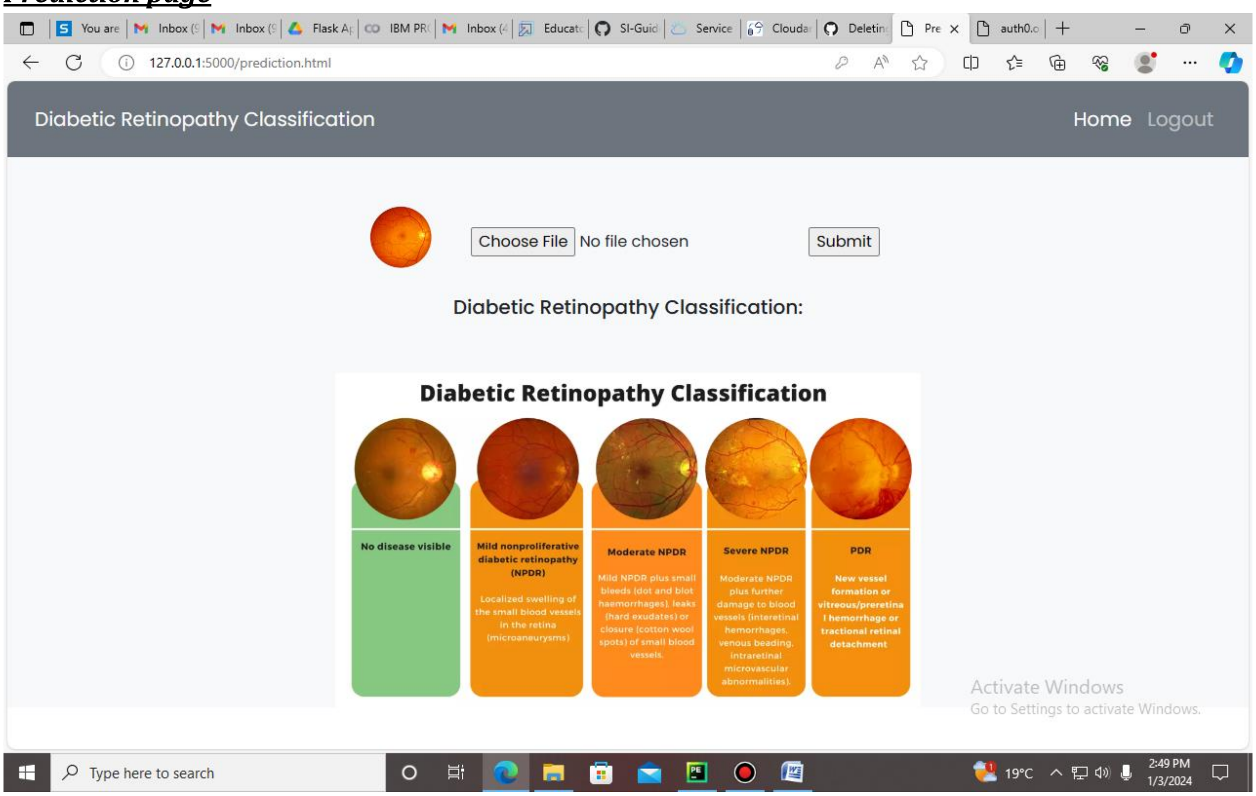
Registeration page

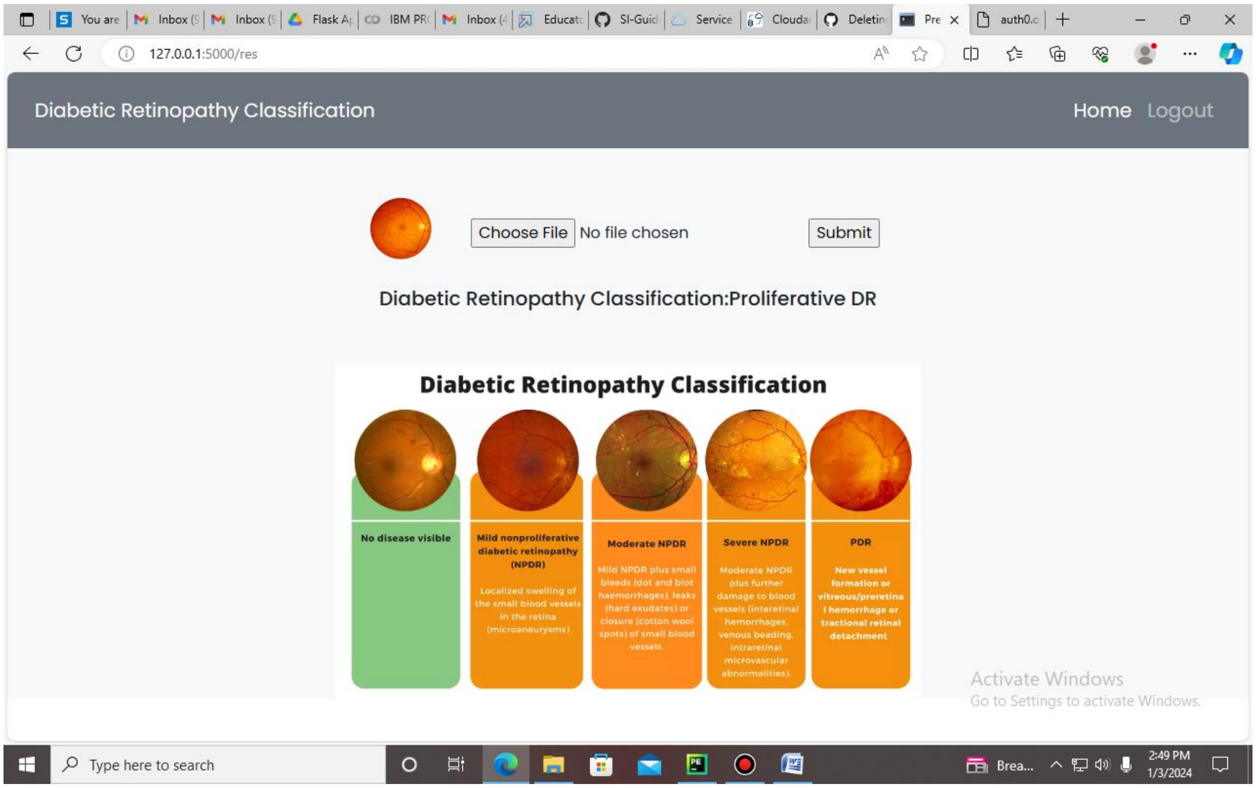


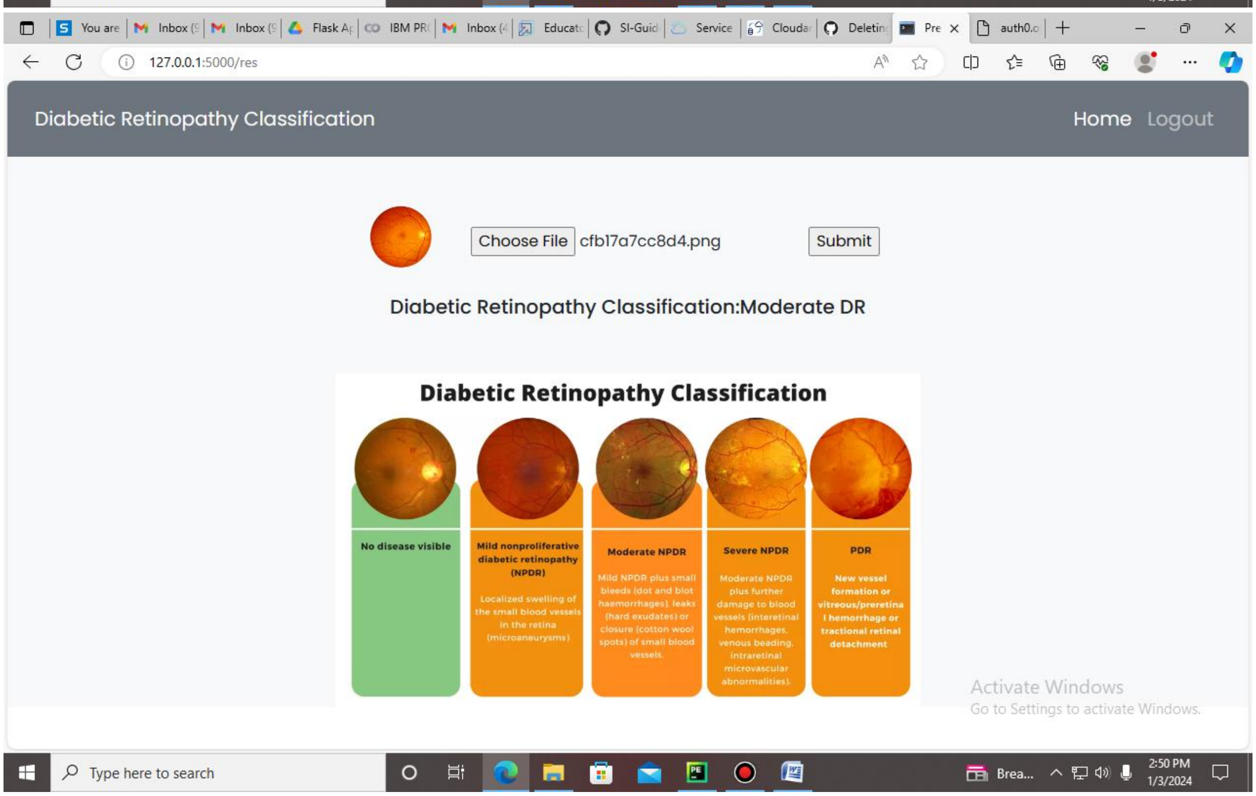
Login page



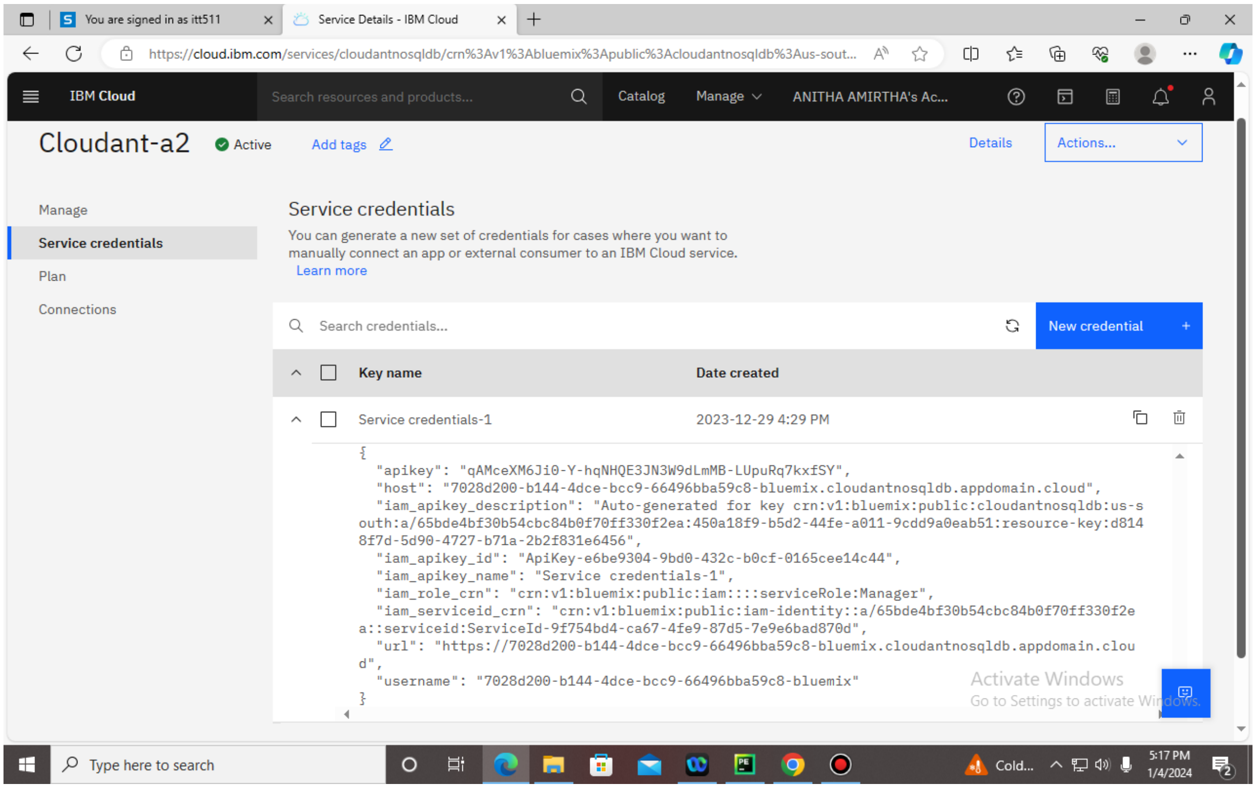
Prediction page

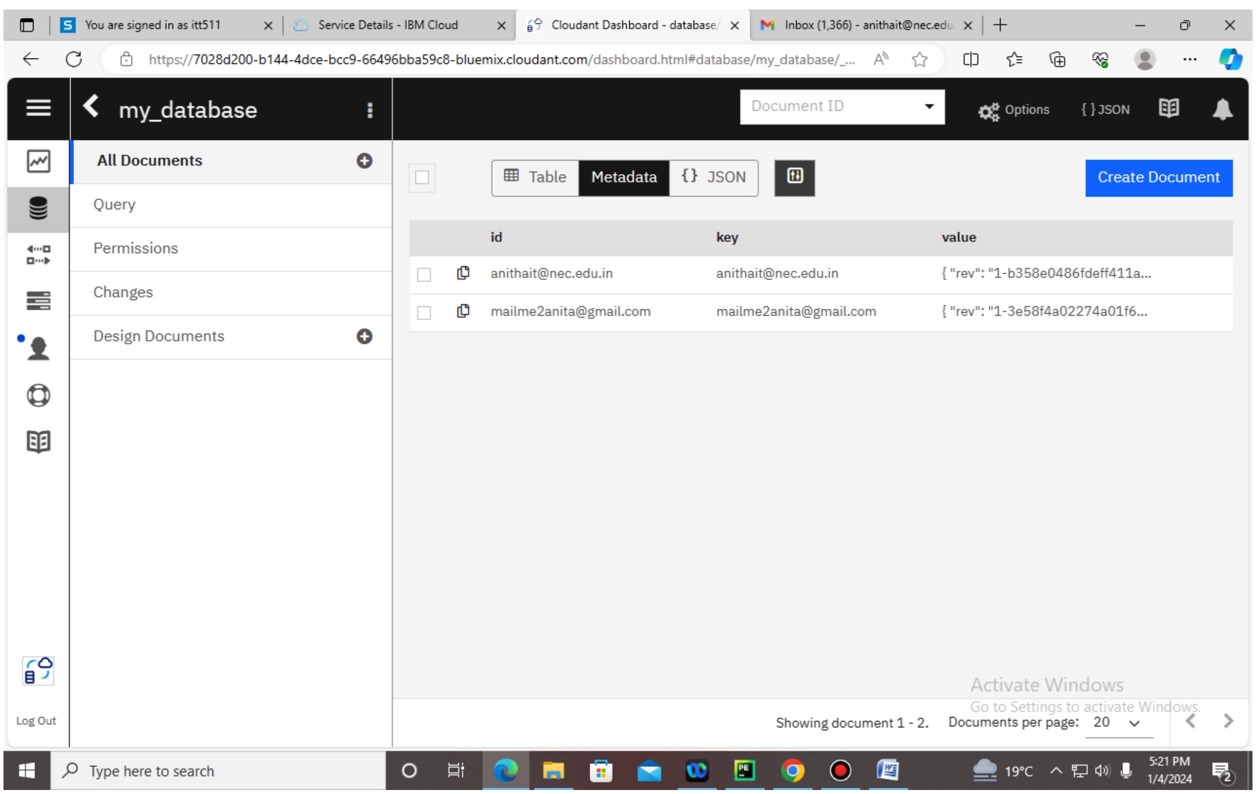



127.0.0.1:5000/res
Diabetic Retinopathy Classification
Home
Logout
Choose File
No file chosen
Submit
Diabetic Retinopathy Classification:Proliferative DR
Diabetic Retinopathy Classification
No disease visible
Mild nonproliferative diabetic retinopathy (NPDR)
Localized swelling of the small blood vessels in the retina (microaneurysms)
Moderate NPDR
Mild NPDR plus small bleeds (dot and blot haemorrhages), leaks (hard exudates) or closure (cotton wool spots) of small blood vessels.
Severe NPDR
Moderate NPDR plus further damage to blood vessels (interetinal hemorrhages, venous beading, intraretinal microvascular abnormalities).
PDR
New vessel formation or vitreous/preretinal hemorrhage or tractional retinal detachment
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:49 PM
1/3/2024


127.0.0.1:5000/res
Diabetic Retinopathy Classification
Home
Logout
Choose File
cfb17a7cc8d4.png
Submit
Diabetic Retinopathy Classification:Moderate DR
Diabetic Retinopathy Classification
No disease visible
Mild nonproliferative diabetic retinopathy (NPDR)
Localized swelling of the small blood vessels in the retina (microaneurysms)
Moderate NPDR
Mild NPDR plus small bleeds (dot and blot haemorrhages), leaks (hard exudates) or closure (cotton wool spots) of small blood vessels.
Severe NPDR
Moderate NPDR plus further damage to blood vessels (interetinal hemorrhages, venous beading, intraretinal microvascular abnormalities).
PDR
New vessel formation or vitreous/preretinal hemorrhage or tractional retinal detachment
Activate Windows
Go to Settings to activate Windows.
Type here to search
2:50 PM
1/3/2024

Cloudant – Database connection





CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The utilization of Deep Learning Fundus Image Analysis for the Early Detection of Diabetic Retinopathy offers a promising strategy to tackle the challenges linked to manual diagnosis and the pressing need for early identification in individuals with diabetes. The developed deep learning model exhibits the capability to autonomously assess retinal fundus images and identify indications of diabetic retinopathy. This has the potential to significantly reduce dependence on manual diagnosis, leading to expedited evaluations and interventions. The model's proficiency in recognizing early signs empowers healthcare practitioners to implement timely interventions and management protocols. The automated system offers efficiency benefits in terms of both time and resources compared to manual diagnosis. The deep learning model showcases its ability to generalize across different cases and various levels of diabetic retinopathy severity. This adaptability is crucial for ensuring robust performance across diverse patient demographics and varying image characteristics.

REFERENCES

1. Mondal, S.S., Mandal, N., Singh, K.K., Singh, A. and Izonin, I., 2022. Edldr: An ensemble deep learning technique for detection and classification of diabetic retinopathy. Diagnostics, 13(1), p.124.
2. Jena, M., Mishra, D., Mishra, S.P. and Mallick, P.K., 2023. A tailored complex medical decision analysis model for diabetic retinopathy classification based on optimized un-supervised feature learning approach. Arabian Journal for Science and Engineering, 48(2), pp.2087-2099.
3. Eminaga, O., Abbas, M., Shen, J., Laurie, M., Brooks, J.D., Liao, J.C. and Rubin, D.L., 2023. PlexusNet: A neural network architectural concept for medical image classification. Computers in Biology and Medicine, 154, p.106594.
4. Zhang, M., Wen, G., Zhong, J., Chen, D., Wang, C., Huang, X. and Zhang, S., 2023. MLP-like model with convolution complex transformation for auxiliary diagnosis through medical images. IEEE Journal of Biomedical and Health Informatics.
5. Rajput, Y., Gaikwad, S., Dhumal, R. and Gaikwad, J., 2023, April. Classification of Non-proliferative Diabetic Retinopathy in Terms of Dark and Bright Lesions Using Multi-layered Perceptron (MLP). In International Conference on Information and Communication Technology for Intelligent Systems (pp. 99-109). Singapore: Springer Nature Singapore.
6. Saranya, P., Pranati, R. and Patro, S.S., 2023. Detection and classification of red lesions from retinal images for diabetic retinopathy detection using deep learning models. Multimedia Tools and Applications, pp.1-21.
7. Solano, A., Dietrich, K.N., Martínez-Sober, M., Barranquero-Cardeñosa, R., Vila-Tomás, J. and Hernández-Cámara, P., 2023. Deep Learning Architectures for Diagnosis of Diabetic Retinopathy. Applied Sciences, 13(7), p.4445.
8. Mohanty, C., Mahapatra, S., Acharya, B., Kokkoras, F., Gerogiannis, V.C., Karamitsos, I. and Kanavos, A., 2023. Using Deep Learning Architectures for Detection and Classification of Diabetic Retinopathy. Sensors, 23(12), p.5726.
9. Birajdar, U., Gadhave, S., Chikodikar, S., Dadhich, S. and Chiwhane, S., 2020, January. Detection and classification of diabetic retinopathy using AlexNet architecture of convolutional neural networks. In Proceeding of International Conference on Computational Science and Applications: ICCSA 2019 (pp. 245-253). Singapore: Springer Singapore.
10. Pan, Y., Liu, J., Cai, Y., Yang, X., Zhang, Z., Long, H., Zhao, K., Yu, X., Zeng, C., Duan, J. and Xiao, P., 2023. Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases. Frontiers in Physiology, 14, p.160.