**Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy**

Prepared By

R.Madhu

Assistant Professor

Department of Information Technology

National Engineering College

Kovilpatti

**ABSTRACT**

The project description provided outlines a comprehensive and ambitious endeavor aimed at leveraging deep learning techniques, specifically convolutional neural networks (CNNs), for the early detection of diabetic retinopathy (DR). The proposed system seeks to overcome the limitations of manual diagnostic processes by developing an automated and standardized approach for analyzing retinal fundus images. Highlighting the significance of DR as a threat to vision health in individuals with diabetes and emphasizing the critical need for early detection to prevent irreversible consequences. Developing a deep learning model to autonomously analyze retinal images and identify subtle signs of DR in its early stages, aiming to provide timely intervention and reduce dependence on subjective manual assessments. Involving training the deep learning model on a diverse dataset of retinal fundus images representing various stages of DR. The model learns intricate patterns and features indicative of early DR manifestations, followed by rigorous validation and testing procedures to ensure reliability and generalizability. Expecting two primary impacts: early detection of DR for timely intervention and the establishment of an efficient, scalable diagnostic solution that surpasses the limitations of manual assessments, potentially revolutionizing DR diagnosis and improving patient outcomes. Highlighting the transformative potential of deep learning in early DR detection, emphasizing the convergence of medical expertise and technological innovation to advance ophthalmic care and healthcare as a whole.

**CHAPTER 1**

**INTRODUCTION**

Retinopathy, particularly diabetic retinopathy (DR), stands as a significant concern within the realm of ophthalmology, posing a substantial threat to the vision health of individuals afflicted by diabetes mellitus worldwide. Among the myriad complications that accompany diabetes, the ocular implications often manifest in the form of retinal damage, leading to vision impairment and, if left unchecked, irreversible blindness. The prevalence of diabetes continues to escalate globally, with projections estimating an upward trajectory in the coming years. Correspondingly, the incidence of diabetic retinopathy serves as a poignant reminder of the critical need for timely and accurate detection, aiming to mitigate its devastating ocular consequences.

* 1. **Diabetic Retinopathy and Its Difficulties to Diagnose**

Diabetes mellitus is a common metabolic disease that is associated with a number of consequences, one of which is DR, which is a major risk factor for visual acuity. Due to the complex pathophysiology of DR, the retina experiences microvascular alterations that result in the creation of recognizable lesions. The present paradigm for diagnosis depends on subjective and variable manual evaluations performed by experienced ophthalmologists, which can delay the timely identification of DR. The shortcomings of the existing diagnosis strategy highlight how urgent it is to look into novel solutions as the prevalence of diabetes rises worldwide.

The diagnosis of DR with the traditional approach mostly depends on the skill of ophthalmologists who manually assess retinal fundus pictures to determine whether or not DR is present and how severe it is. Although there is no doubting the expertise of these medical professionals, the procedure is time-consuming, resource-intensive, and susceptible to inter-observer variability. There is a huge gap in the supply of qualified ophthalmologists, which causes delays in diagnosis and treatment.

Furthermore, there is some degree of heterogeneity in the assessment of DR due to the subjective nature of manual grading. Inter-observer differences, impacted by things like weariness and experience, can lead to incorrect diagnoses and affect how consistently patients are treated. The increasing global prevalence of diabetes raises urgent concerns about the scalability of the existing diagnostic technique, hence calling for creative solutions that might enhance and accelerate the diagnostic process.

Diabetic Retinopathy (DR) is now diagnosed primarily through manual examinations performed by ophthalmologists. This approach has a number of drawbacks and difficulties that reduce the diagnostic process' efficacy and efficiency.

* 1. **Deep Learning in Medical Imaging**

Deep Learning has proven very effective in image processing applications, especially when using convolutional neural networks (CNNs). It is a perfect fit for the detailed analysis needed in medical imaging because of its capacity to identify intricate patterns and features within big datasets. Deep learning has the potential to improve diagnostic precision and facilitate prompt intervention in the context of depression and anxiety (DR), when early indicators may be imperceptible. Deep learning can overcome the drawbacks of manual diagnosis by automating the processing of fundus images, offering a standardized and scalable method.

A subset of artificial intelligence (AI) called deep learning (DL) has become a game-changing paradigm in a number of industries, medical imaging being one of the most promising. The use of deep learning (DL) in medical imaging, such as the examination of retinal fundus pictures for diseases such as diabetic retinopathy (DR), has enormous potential to revolutionize current diagnostic methods. Deep learning holds great promise for medical imaging for a number of reasons:

1. **Pattern Recognition and Feature Extraction:**

Convolutional Neural Networks (CNNs), in particular, are deep learning algorithms that are particularly good at extracting pertinent information and learning nuanced patterns from large, complicated datasets. Deep learning models' capacity to recognize and understand these aspects is crucial in medical imaging, where minute details and patterns may have diagnostic importance. This skill is especially important when it comes to disorders like DR, where early symptoms can be difficult to identify and require a high degree of sensitivity and specificity.

1. **Acquiring Knowledge from Datasets:**

Big, varied datasets are ideal for deep learning models to perform well. These models can adapt well to new, unseen data because they may learn from a multitude of photos that depict different stages of a medical disease. A deep learning model trained on a broad dataset can successfully represent the range of retinal alterations associated with different phases of the condition. This is especially useful in the case of DR, since the disease progression extends from mild to severe stages.

1. **Standardization and Automation:**

Robust image analysis activities can be automated with deep learning, lowering the need for human subjectivity in manual evaluations and, ultimately, the unpredictability they introduce. An impartial and uniform assessment of medical photographs is guaranteed by the standardized methodology made possible by deep learning models. This is especially important when diagnosing drug-resistant strains (DRs), as accurate grading and early detection are essential for prompt treatment.

1. **Efficiency and Scalability:**

Deep learning models' high computational efficiency makes it possible to analyze a lot of medical photos quickly. This effectiveness is essential for overcoming the scaling issues that conventional manual diagnostic techniques encounter. The capacity to process a large number of images effectively is critical to improve access to early detection in the setting of DR, where the demand for timely screenings outpaces the supply of qualified eye care practitioners.

1. **Adaptability and Continuous Learning:**

Deep learning models are capable of improving their performance over time as a result of being exposed to fresh data since they demonstrate adaptability and continuous learning. Deep learning models are highly adaptable, which guarantees their continued relevance and effectiveness in the face of new difficulties in the ever-changing field of medical imaging.

* 1. **Objective of the Project**

This project's main goal is to create a Deep Learning Fundus Image Analysis system that can recognize early symptoms of DR on its own. By utilizing a broad range of datasets that span different phases of the illness, the deep learning model seeks to identify complex patterns characteristic of drug resistance (DR), providing a more uniform and impartial method of diagnosis. This work aims to solve important problems like resource intensity and inter-observer variability related to manual assessments, in addition to pushing the boundaries of technology.

**CHAPTER 2**

**LITERATURE REVIEW**

This chapter discusses about the various methodologies used to perform the image analysis of early detection of diabetic retinopathy.

* 1. **Residual Neural Networks (ResNet)**

The problems associated with training extremely deep networks were addressed by ResNet, a kind of CNN that introduced residual learning. Fundus image analysis has been used to collect complex information related to diabetic retinopathy using ResNet topologies. The introduction of residual blocks is the main innovation of ResNet. A shortcut link that skips one or more levels, sometimes referred to as an identity connection or skip, makes up a residual block. ResNet learns the residual mapping, or the difference between the input and output, as opposed to the direct mapping. The total of the input and the learned residual is the output of a residual block.

* 1. **Inception Networks (GoogLeNet)**

DR detection has made use of inception networks, which are distinguished by their inception modules. These structures are useful for assessing retinal images with different sizes of lesions since they are made to capture multi-scale characteristics. The core architectural element of GoogLeNet is the inception module. Inception modules use several filter sizes (1x1, 3x3, 5x5) and parallel pooling processes instead of depending on a single receptive field size. This lets the network learn a wide range of features at various scales by allowing it to simultaneously record coarser and finer features. Before applying larger convolutions, Inception modules frequently employ 1x1 convolutions to minimize the dimensionality of input feature maps. The two main roles of these 1x1 convolutions are dimension reduction and the introduction of non-linearity via activation functions.

* 1. **Capsule Networks (CapsNets)**

As a substitute for conventional CNNs, CapsNets concentrate on capturing the hierarchical relationships between features. In challenges involving the comprehension of spatial hierarchies in medical images, such as fundus images for DR diagnosis, they have demonstrated promise. Capsules are the basic units of construction in CapsNets. Capsules, in contrast to neurons in conventional networks, are made to encode several qualities, including orientation, scale, and position, and to represent entities in the incoming data. In comparison to conventional neural network units, capsules cooperate to contain the instantiation parameters of certain entities, offering a more sophisticated representation.

* 1. **Generative Adversarial Networks (GANs):**

The use of GANs to enhance already-existing datasets or produce artificial retinal images has been investigated. They can help train deep learning models by offering a variety of retinal abnormality examples. A neural network called the generator is in charge of creating fresh data samples. Its input is random noise, which it converts into data that should ideally be identical to real data. The generator aims to produce data that fools the discriminator in a realistic enough way.

An additional neural network called the discriminator assesses the veracity of a given data sample. Its function is to discriminate between data that is generated by the generator and actual data. With proper training, the discriminator can accurately identify samples as authentic or fraudulent. The discriminator and generator engage in constant interaction during the training process. While the discriminator aims to become more skilled at differentiating real from fake, the generator works to enhance its capacity to produce realistic data. A feedback loop produced by this adversarial training makes both networks better.

GANs' goal is presented as a minimax game. While the discriminator tries to increase its accuracy in differentiating actual from generated samples, the generator tries to decrease the possibility that the discriminator would accurately categorize generated samples as fake. Different loss functions are used by the generator and discriminator. The probability that the discriminator will classify the generated samples as real determines the generator's loss. The probability that the discriminator will classify the generated samples as real determines the generator's loss. The sum of the discriminator's loss and the mistakes made in identifying phony and authentic samples.

The generator and discriminator are iteratively changed during training. The discriminator assesses the fresh samples produced by the generator. The generator is then updated to produce more persuasive examples by using the gradients from the discriminator's evaluation. GANs are vulnerable to a problem called mode collapse, in which certain modes in the data distribution are ignored by the generator, which results in a restricted range of samples. To solve this problem, researchers have developed methods like spectral normalization and minibatch discriminating. Numerous advancements and expansions have been made to GANs, such as Conditional GANs (cGANs), which produce samples contingent on particular input data. Applications for GANs include text-to-image synthesis, super-resolution, picture-to-image translation, image production, and style transfer.

**CHAPTER 3**

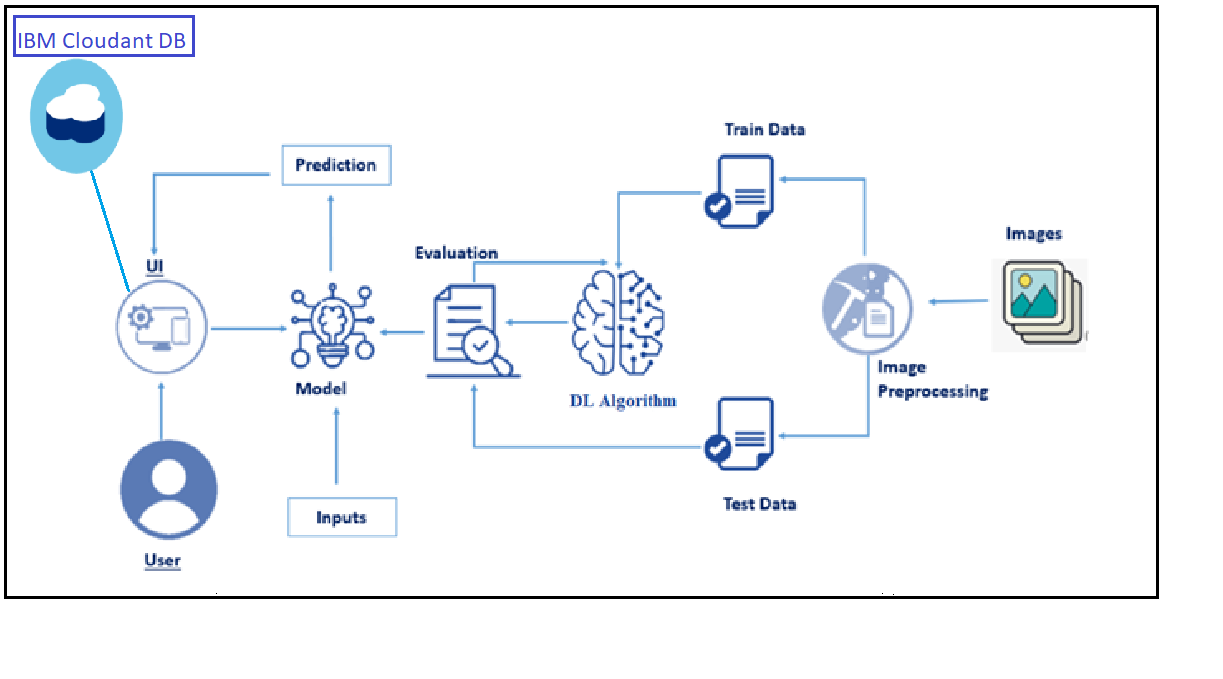
**SYSTEM DESIGN**

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

**3.1 Proposed System**

The flow of the proposed system was shown in Figure 3.1. The steps consists of

* Data Collection
* Image Preprocessing
* Training and Testing
* Model building
* Performance analysis
* Prediction using GUI



**Fig 3.1 Architecture of proposed system**

To prepare the training and testing portions, data was gathered from the dataset and processed. With SoftMax as the activation function, the model is built using Xception Net. Next, the Python Flask application's testing photos were used to make the prediction. The accuracy and loss metrics are used to compare the model's performance.

* 1. **Data collection**

Building machine learning models requires gathering data, especially for initiatives like the diagnosis of diabetic retinopathy. The purpose of data collection in this project is to compile a dataset of retinal fundus images, which are necessary for the deep learning model's training and testing.

**3.3 Image Preprocessing**

An essential first step in getting raw retinal fundus images ready for deep learning model training is image preparation. An increased model's capacity to learn pertinent features and patterns is facilitated by proper preprocessing. The following picture preprocessing methods are frequently used when detecting diabetic retinopathy:

Adjust the picture sizes to a standard resolution. Standardizing the image dimensions helps with efficient processing during model training and guarantees consistency throughout the dataset. Set the values of the pixels to a standard scale, usually between 0 and 1. Normalization guarantees that the model is not sensitive to changes in pixel intensity and aids in stabilizing the training process. Images should be cropped to highlight pertinent areas of interest, like the macula and optic disc. In addition to ensuring that the model concentrates on crucial regions for the diagnosis of diabetic retinopathy, this lowers computational complexity.

**3.4 Training and Testing**

In order to create a machine learning model for the diagnosis of diabetic retinopathy, training and testing are essential stages. In these stages, the model is trained on a labeled dataset, and its performance is assessed on an additional set of data. Separate the dataset into a testing set and a training set. The testing set assesses the model's performance on unobserved data, while the training set is used to train the model.

**3.5 Model building**

Selecting the pre-trained Xception model, adding more layers, and setting the learning process are the stages involved in developing a model for diabetic retinopathy diagnosis using the Xception architecture. The Keras library, a high-level deep learning API that operates on top of TensorFlow, will be used in this example.

A deep convolutional neural network (CNN) architecture with impressive results in image-related tasks is the Xception model. Its ability to capture complex hierarchical information is attributed to its great depth and the usage of depthwise separable convolutions.

* **Loading the Pre-trained Xception Model**
* **Freezing the Pre-trained Layers**
* **Custom Model Architecture**
* **Global Average Pooling Layer**
* **Dense Layer with Softmax Activation**
  1. **Performance analysis**

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.

(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. For classification tasks, categorical cross entropy is commonly used. For regression tasks, mean squared error (MSE) might be used. Lower loss values indicate better alignment between predicted and true values. Monitoring the loss during training helps in understanding the convergence of the model. A decreasing loss over epochs suggests that the model is learning from the data.

* 1. **Prediction using GUI**

After the model building, testing images are undergone following steps to idntify the level of diabetic retinopathy.

* Load Testing Images
* Preprocess Testing Images
* Model Prediction

Ensure that the testing images are preprocessed in a manner consistent with the preprocessing applied during training.Evaluate the model's performance using appropriate metrics based on the task (accuracy, precision, recall, F1 score, etc.).Keep in mind any specific requirements or constraints of your application or project when interpreting and using the model predictions.For a multi-class classification task, consider using the argmax function to get the predicted class index

**CHAPTER 4**

**IMPLEMENTATION**

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

**4.1 Structure of implementation**

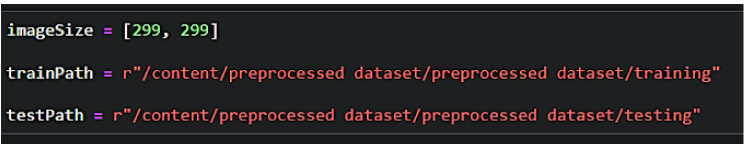
The proposed methodology implemented based on the following flow,

* Download the dataset
* Model Building
* Cloudant DB
* Application Building

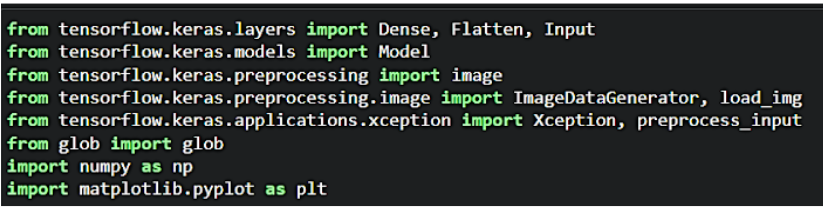
Dataset downloaded from Kaggle, and images are preprocessed, testing and training images are separated. Model built using Xception and Softmax. CloudantDB created then Python Flask application used to build web application.

**4.2 Dataset**

To build a DL model we have to split training and testing data into two separate folders. But In the project dataset folder training and testing folders are presented. So, in this case have to assign a variable and pass the folder path to it. Four different transfer learning models are used in our project and the best model (Xception) is selected.The image input size of xception model is 299, 299.



Import the necessary libraries as shown in the image.

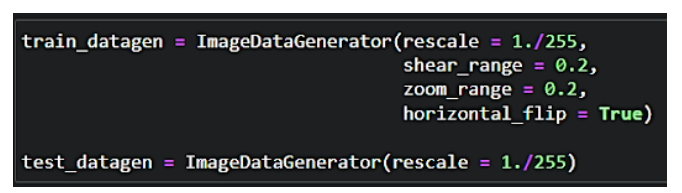


ImageDataGenerator class is instantiated and the configuration for the types of data augmentation

There are five main types of data augmentation techniques for image data; specifically:

* Image shifts via the width\_shift\_range and height\_shift\_range arguments.
* The image flips via the horizontal\_flip and vertical\_flip arguments.
* Image rotations via the rotation\_range argument
* Image brightness via the brightness\_range argument.
* Images zoom via the zoom\_range argument.

An instance of the ImageDataGenerator class can be constructed for train and test.

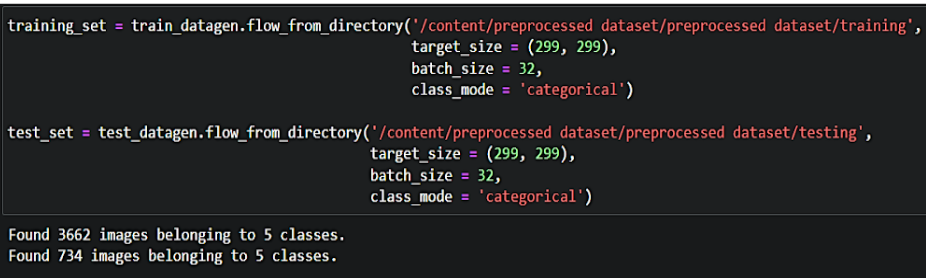


To apply ImageDataGenerator functionality to the Train set and Test set by using the following code. For Training set using flow\_from\_directory function.

This function will return batches of images from the subdirectories

Arguments:

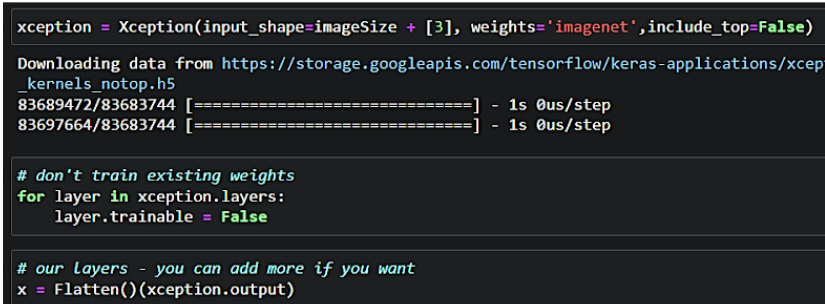
* Directory: Directory where the data is located. If labels are "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
* batch\_size: Size of the batches of data which is  64.
* target\_size: Size to resize images after they are read from disk.
* class\_mode:



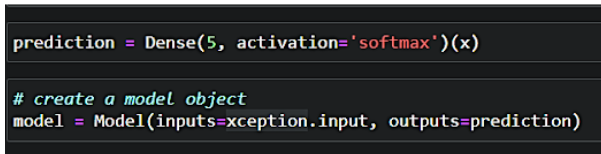
**4.3 Model Building**

For one of the models, used it as a simple feature extractor by freezing all the five convolution blocks to make sure their weights don’t get updated after each epoch as we train our own model. Here, we have considered images of dimension (229, 229, and 3). Also, assigned include\_top = False because we are using convolution layer for features extraction and wants to train fully connected layer for our images classification(since it is not the part of Imagenet dataset)

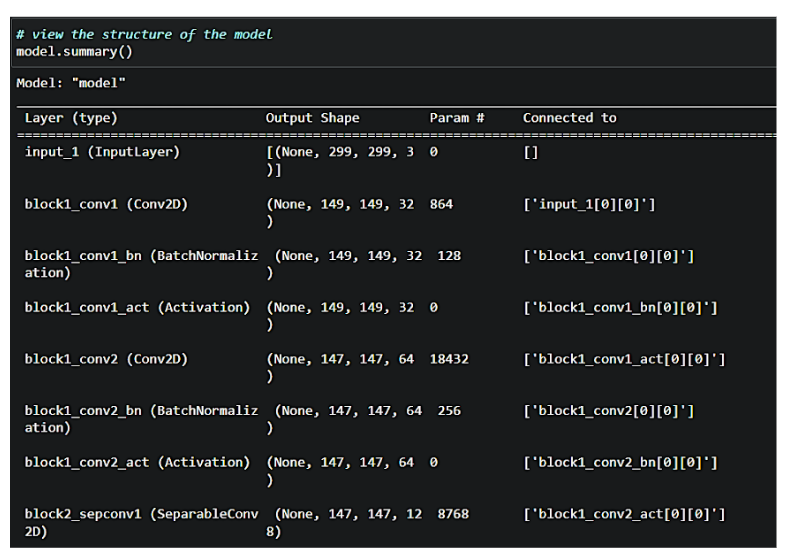
Flatten layer flattens the input. Does not affect the batch size



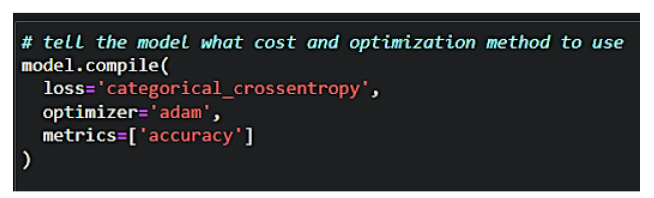
A dense layer is a deeply connected neural network layer. It is the most common and frequently used layer. Create a model object named model with inputs as xception.input and output as dense layer**.**



The number of neurons in the Dense layer is the same as the number of classes in the training set. The neurons in the last Dense layer, use softmax activation to convert their outputs into respective probabilities.  Understanding the model is a very important phase to properly use it for training and prediction purposes. Keras provides a simple method, summary to get the full information about the model and its layers.



The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires a loss function during the model compilation process.Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here by using adam optimizerMetrics are used to evaluate the performance of your model. It is similar to the loss function, but not used in the training process.

 Now, train the model with our image dataset. The model is trained for 30 epochs and after every epoch, the current model state is saved if the model has the least loss encountered till that time. We can see that the training loss decreases in almost every epoch till 10 epochs and probably there is further scope to improve the model.

**fit\_generator** functions used to train a deep learning neural network

**Arguments:**

* steps\_per\_epoch: it specifies the total number of steps taken from the generator as soon as one epoch is finished and the next epoch has started. We can calculate the value of     steps\_per\_epoch as the total number of samples in your dataset divided by the batch size.
* Epochs: an integer and number of epochs we want to train our model for.
* validation\_data can be either:

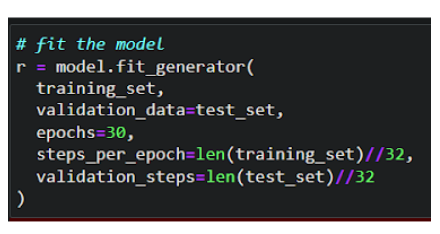
                      - an inputs and targets list

                      - a generator

                      - an inputs, targets, and sample\_weights list which can be used to evaluate

                        the loss and metrics for any model after any epoch has ended.

* validation\_steps: only if the validation\_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.



The model is saved with .h5 extension as follows An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.



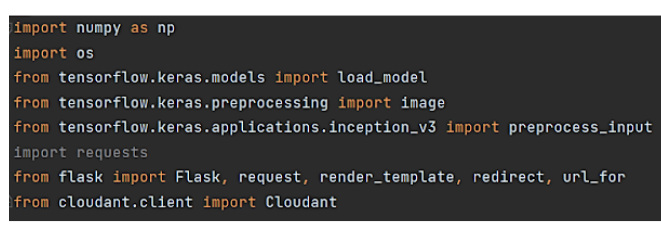
**4.3 Cloudant DB**

* Log in to IBM Cloud account, and click on Catalog
* To create the connection information that your application needs to connect to the instance, click New credential.
* Enter a name for the new credential in the Add new credential window.
* Accept the Manager role.
* (Optional) Create a service ID or have one automatically generated for you.
* (Optional) Add inline configuration parameters. This parameter isn't used by IBM Cloudant service credentials, so ignore it.
* Click Add.
* To see the credentials that are required to access the service, click the chevron.
* In order to manage a connection from a local system you must first initialize the connection by constructing a Cloudant client.We need to import the cloudant library.
* IBM Cloud Identity & Access Management enables you to securely authenticate users and control access to all cloud resources consistently in the IBM Bluemix Cloud Platform.

**4.4 Application Building**

In python flask HTML files, CSS and required images are placed in temlpate and static folder. The main application is created by the following steps,

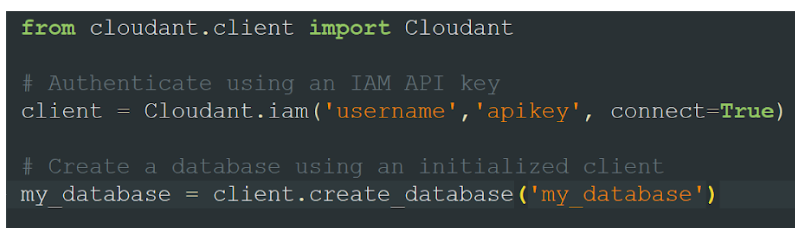
* Import Libraries
* Create Database
* Render HTML pages
* Configure the separate pages
* Predcition on UI



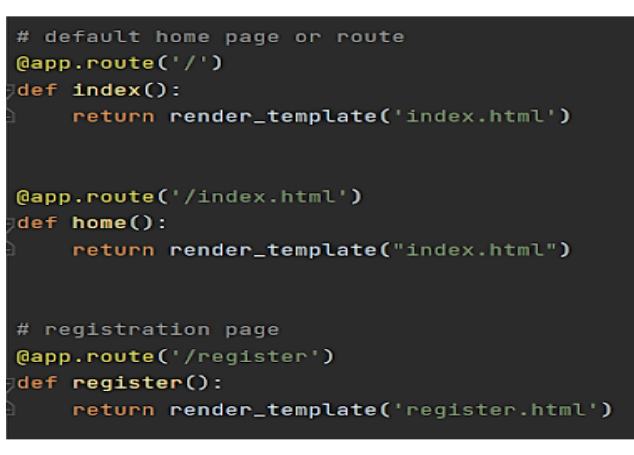
Load the saved model. Importing the flask module in the project is mandatory. An object of Flask class is our WSGI application. Flask constructor takes the name of the current module (\_\_name\_\_) as argument.



Create a database using an initiated client.



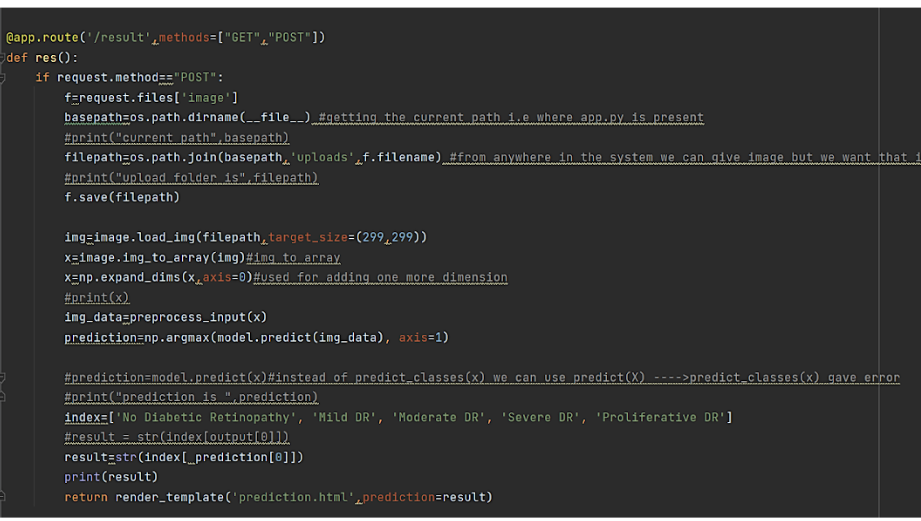
Render HTML page



Based on user input into the registration form stored it on data dictionary then we can validate the data using \_id parameter with user input that can store it on query variable then can validate by passing the query variable into the my\_database.get\_user\_result() method.Then can check the docs length by using len(docs.all()) function.If the length of docs is 0 then user will register successfully on the platform and user data will store on the database.Otherwise its shows the message as user already registered please login and use web application for DR prediction. Based on user input into the login form we stored user id and password into the (user,passw) variables. Then validate the credentials using \_id parameter with user input that store it on query variable then we can validate by passing the query variable into the my\_database.get\_user\_result() method.Then we can check the docs length by using len(docs.all()) function.If the length of doc is 0 then it means username is not found.Otherwise its validate the  data that is stored on the database and check the username & password. If it's matched then the user will be able to login and use web application for DR prediction.Otherwise the user needs to provide correct credentials.



The image is selected from uploads folder. Image is loaded and resized with load\_img() method. To convert image to an array, img\_to\_array() method is used and dimensions are increased with expand\_dims() method. Input is processed for xception model and predict() method is used to predict the probability of classes. To find the max probability np.argmax is used.



**CHAPTER 5**

**RESULTS AND DISCUSSION**

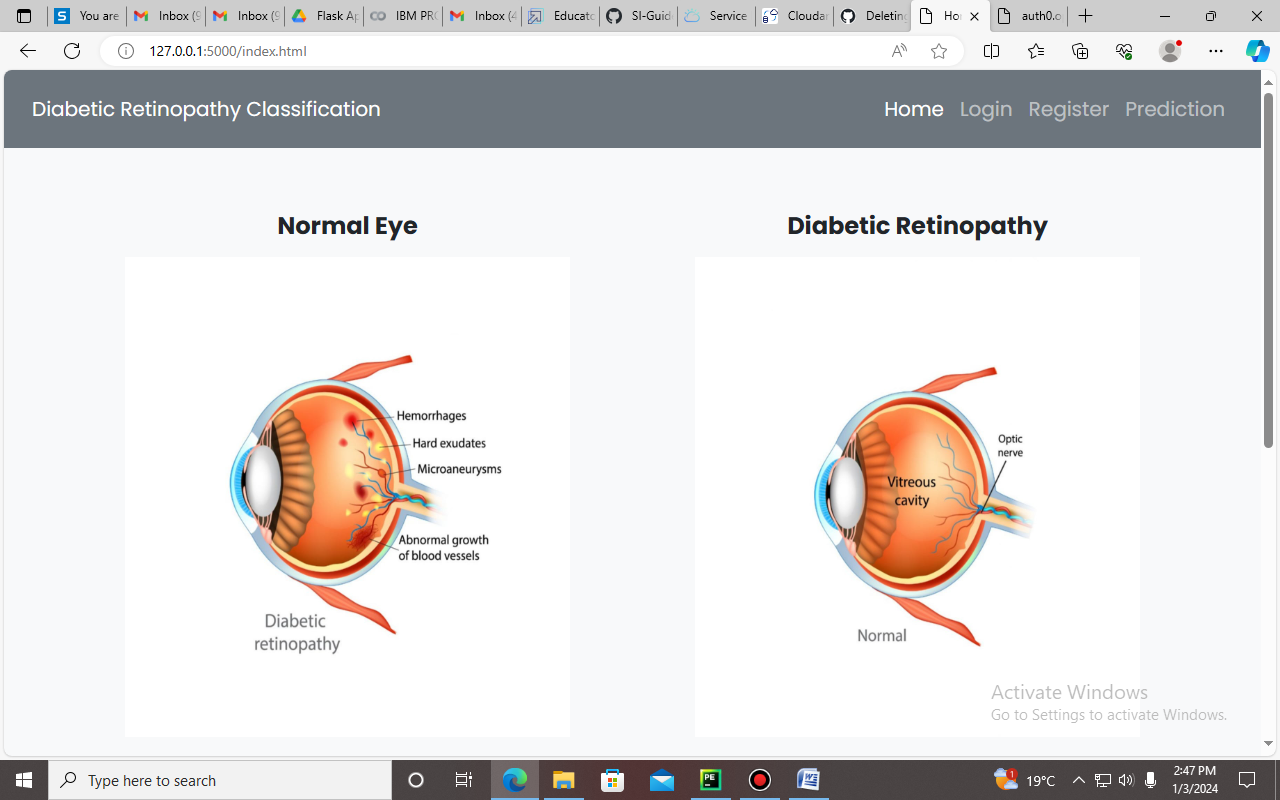
This chapter discusses about the results of the project with prediction by Xception Net using Python Flask application.

**5.1 Web pages**

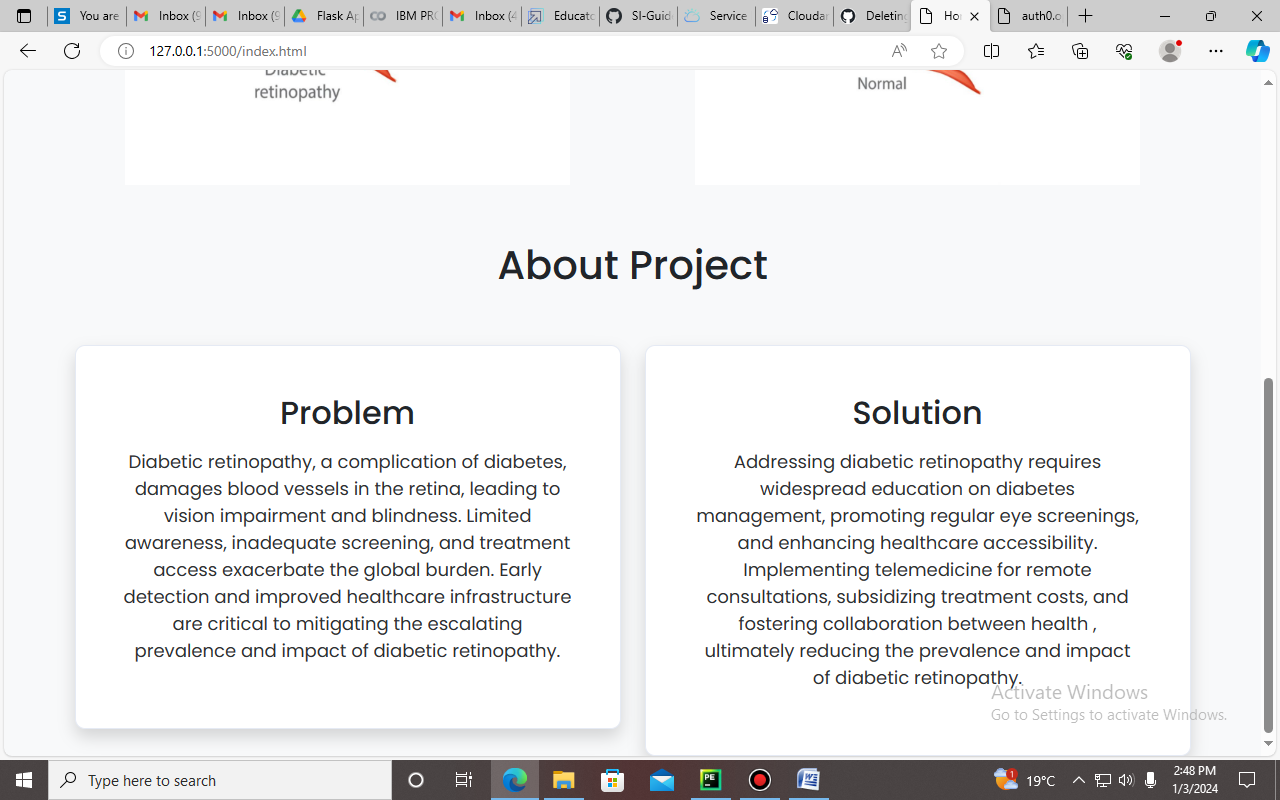
The application consists of following html pages,

* Index page
* Login page
* Register page
* Prediction page
* Logout page

Index page was shown in figure 5.1 and figure 5.2.

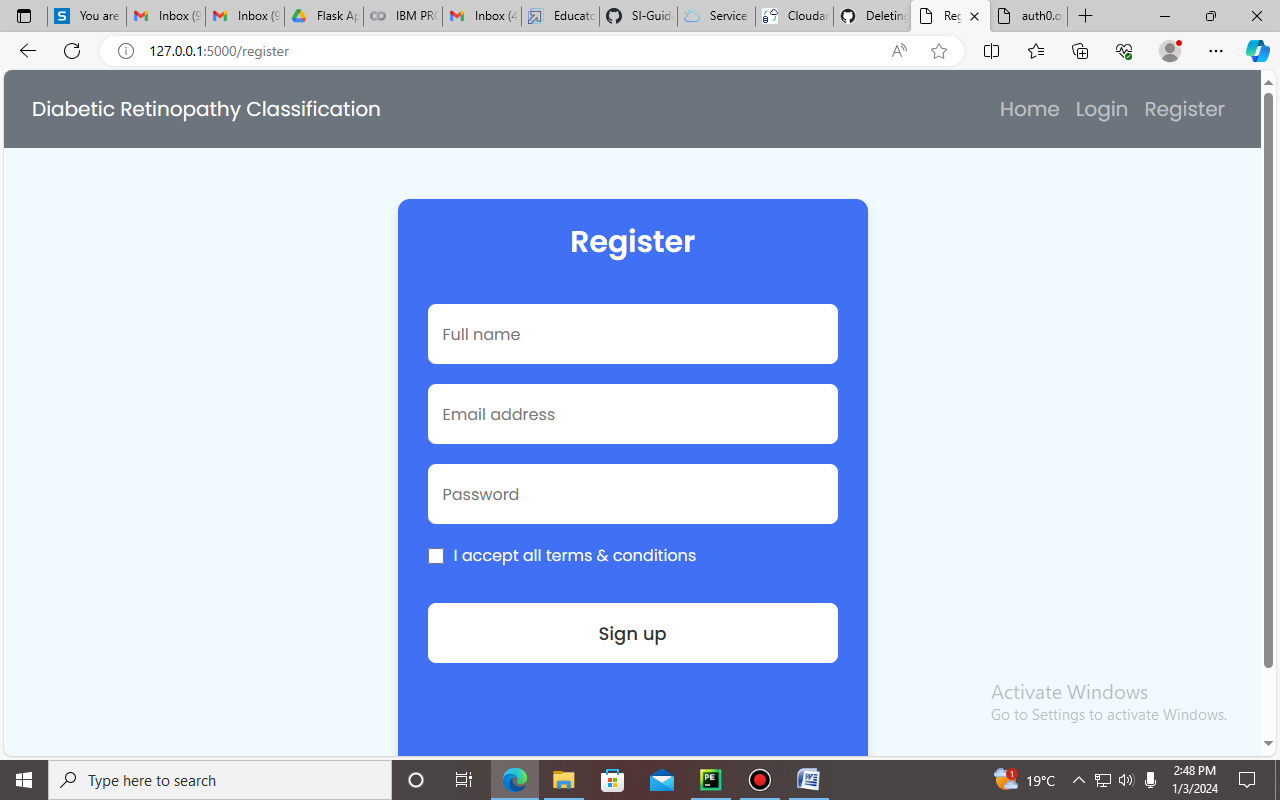


**Fig 5.1 Index Page**



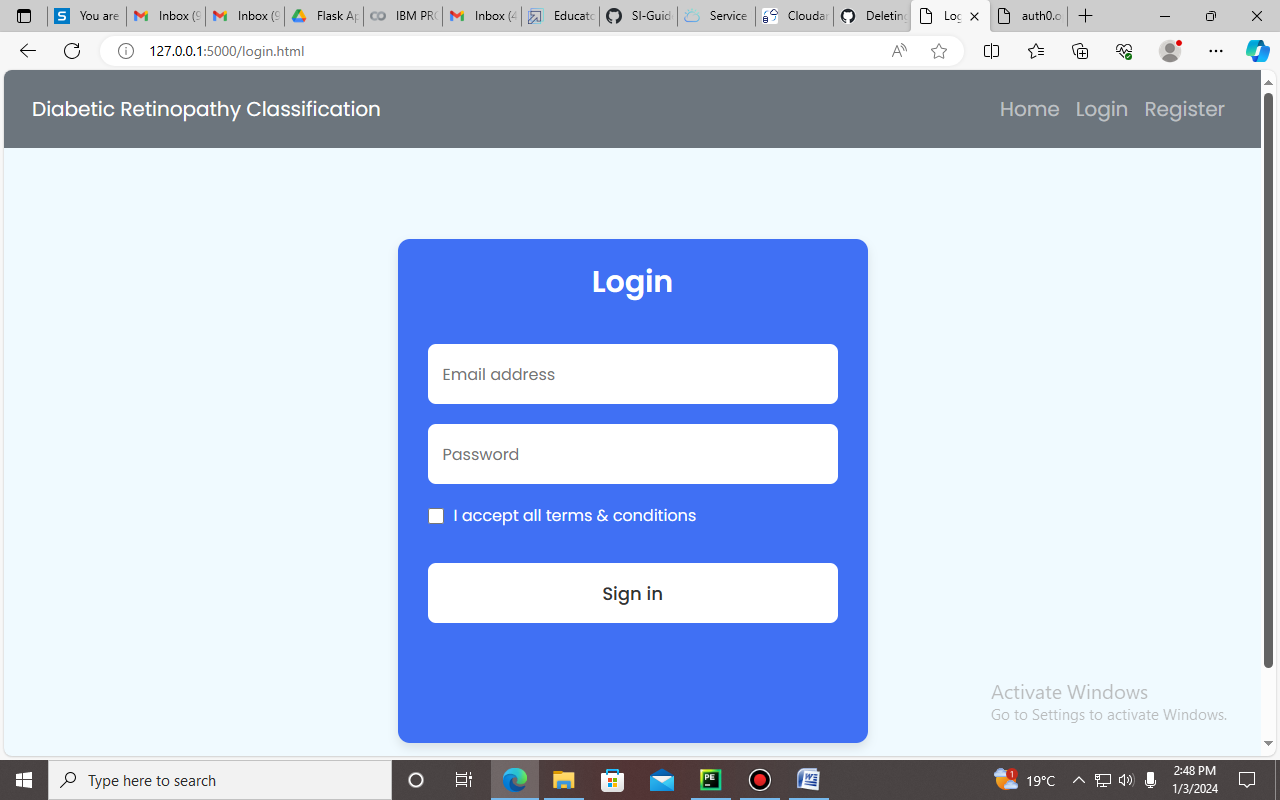
**Fig 5.1 Index Page**

Layout of the Register page was shown in figure 5.3.



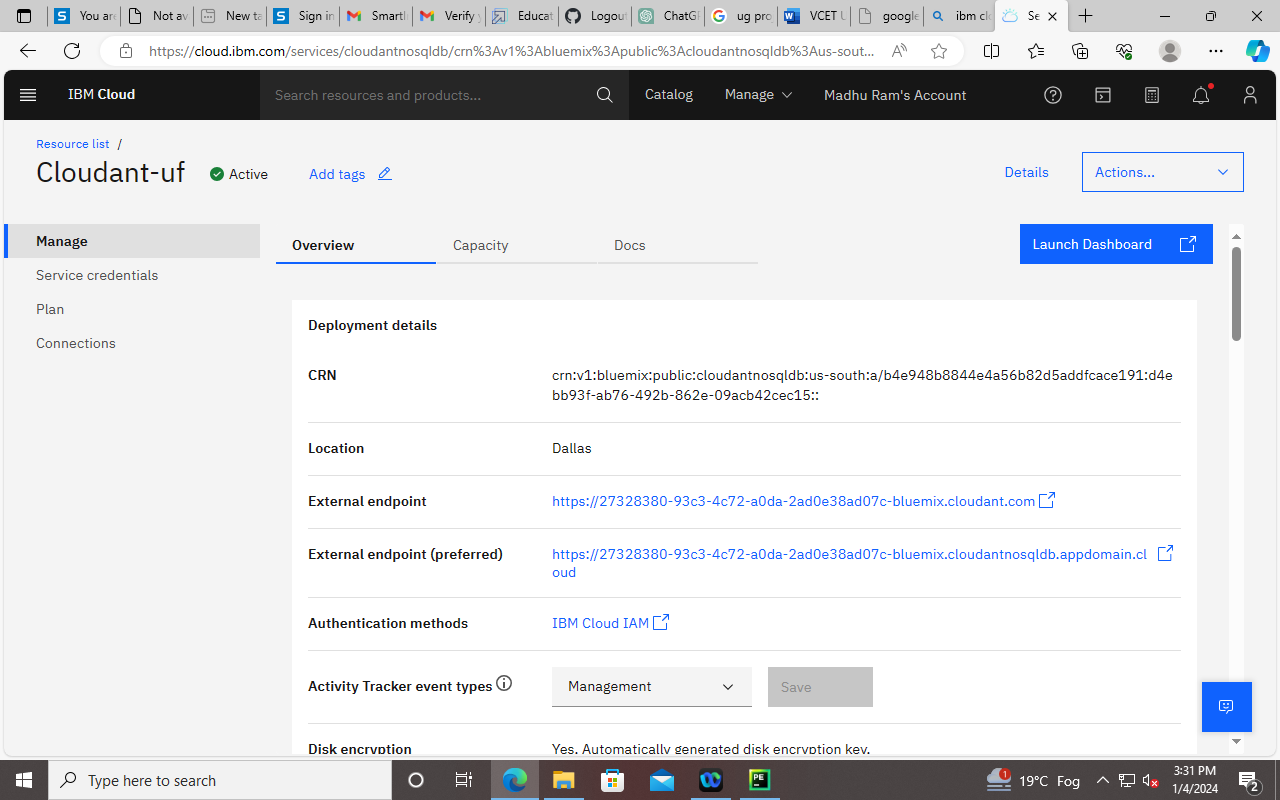
**Fig 5.3 Register Page**

Layout of the Login page was shown in figure 5.4.

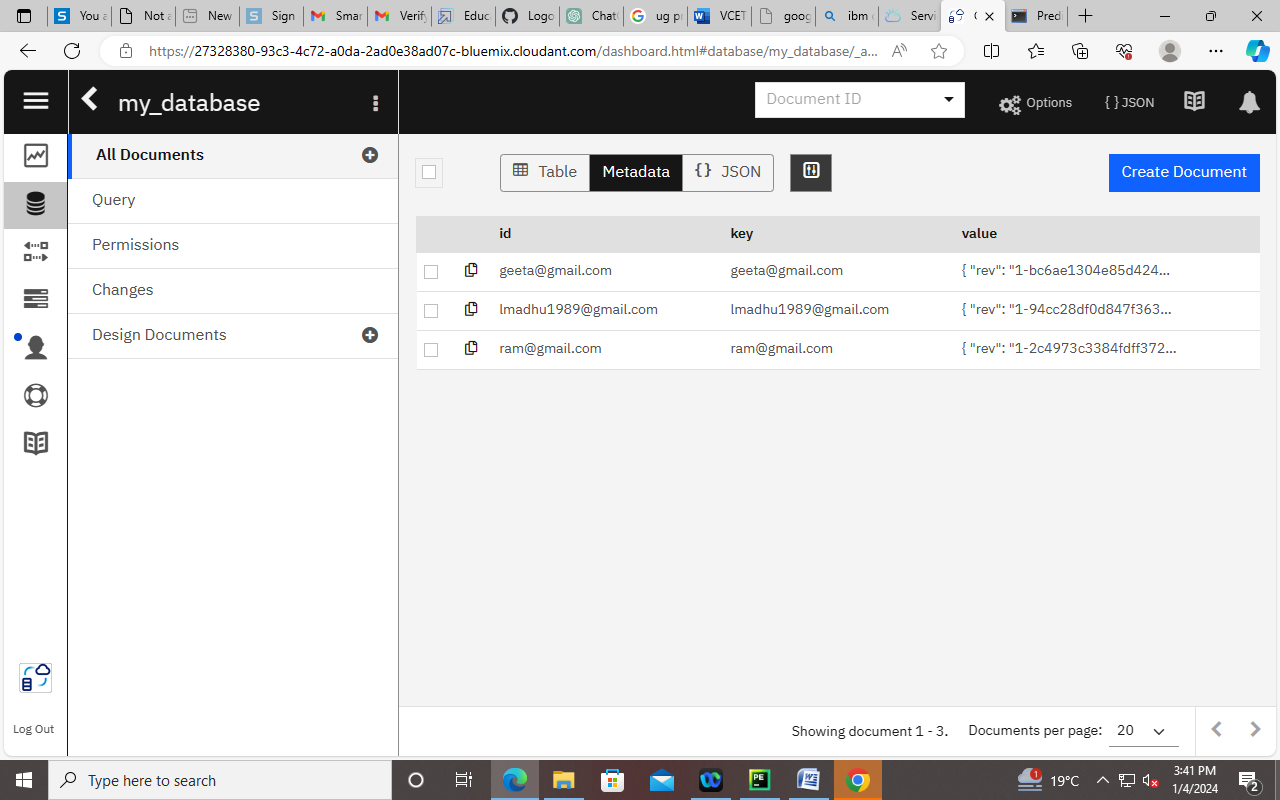


**Fig 5.4 Login Page**

After registration, the details are stroed in CloudantDB. Connection was shown in figure 5.5.

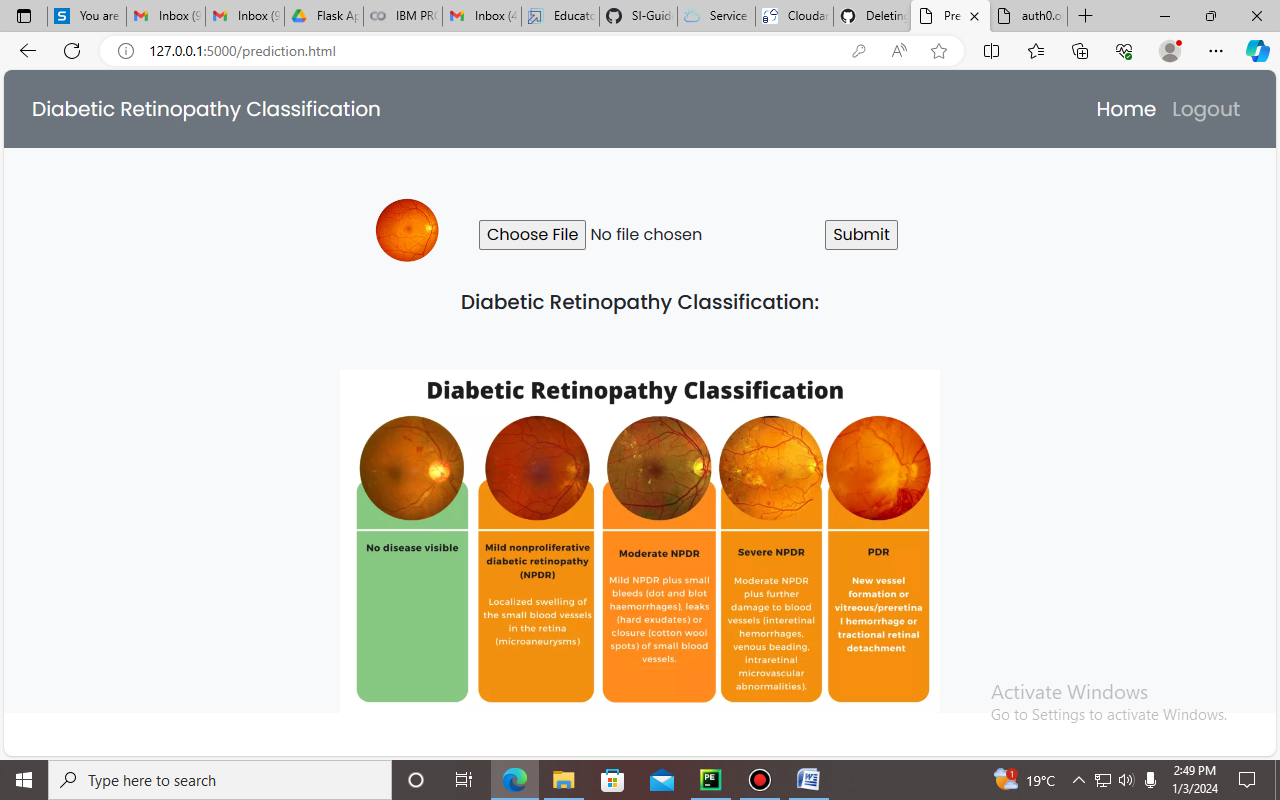


**Fig 5.5 Cloudant DB creation**



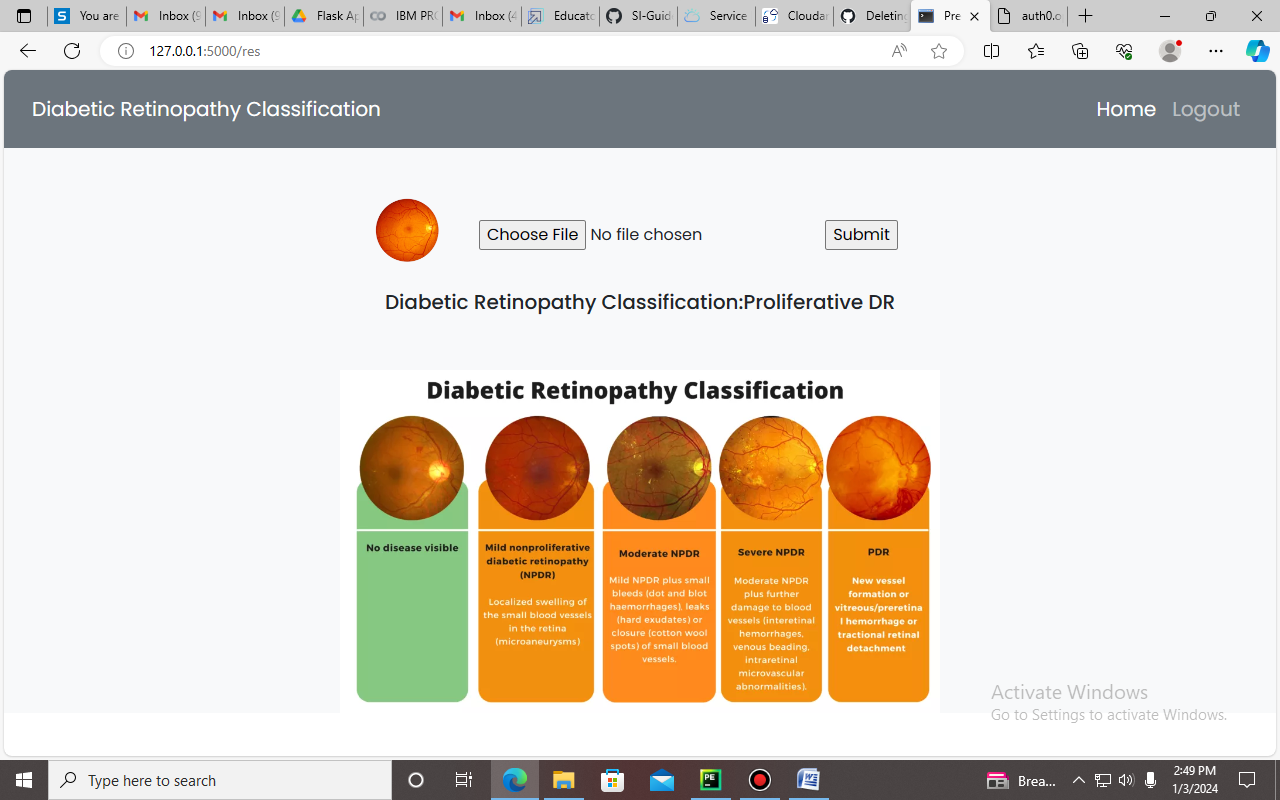
**Fig 5.6 Insertion of details from registration page**

After login prediction done in page shown in figure 5.7.

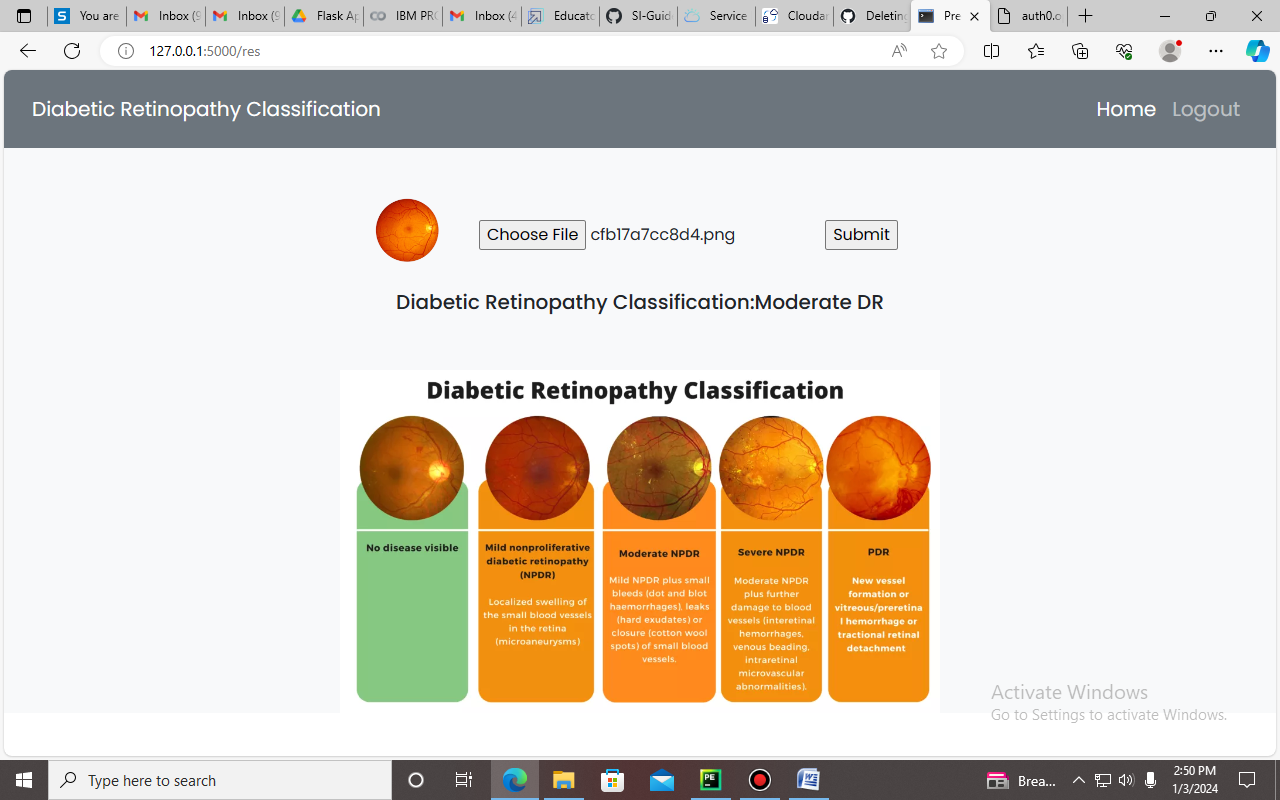


**Fig 5.7 Prediction page layout**

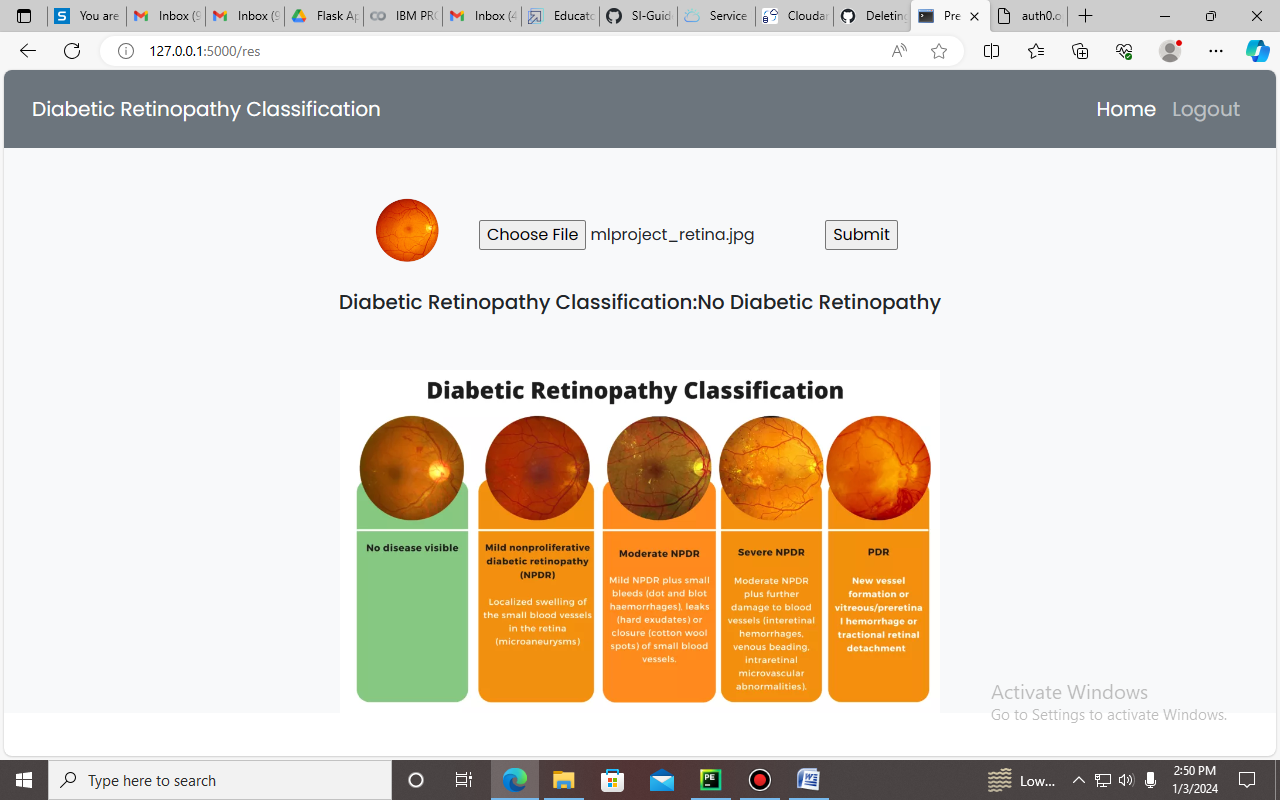
Testing image is loaded in the choose file option, then the predicted class is shown below. Different kinds of prediction was shown in figure 5.8, 5.9,5.10.



**Fig 5.8 Classification of Proliferative DR**



**Fig 5.9 Classification of Moderate DR**



**Fig 5.10 Classification of No Diabetic Retinopathy**

**CHAPTER 6**

**CONCLUSION**

A potential solution to the problems with manual diagnosis and the urgent need for early detection in diabetic patients is the Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy. In order to create an automated and effective solution for diabetic retinopathy detection, the project makes use of cutting-edge deep learning techniques, with a particular focus on the processing of retinal fundus images. The deep learning model that was constructed demonstrates the ability to automatically assess images of the retinal fundus and identify indicators of diabetic retinopathy. This can result in assessments and interventions being completed more quickly by greatly reducing the need for manual diagnosis. It is imperative to identify diabetic retinopathy early in order to stop permanent damage and vision loss. Healthcare personnel are empowered to undertake timely interventions and management measures due to the model's capacity to spot early indications. When compared to manual diagnosis, the automated system is more efficient in terms of time and resources. It can be used in a variety of healthcare settings with a wide range of patient populations since it can be scaled to accommodate enormous datasets. The hazards connected with human error in the manual diagnosis procedure are reduced by the computer-aided diagnosis system. The methodology helps to reduce misdiagnosis by offering a consistent and impartial analysis. The deep learning model exhibits the ability to generalize across a range of cases and severity levels of diabetic retinopathy. Robust performance across various patient demographics and image attributes is contingent upon this.

**REFERENCES**

1. Ramachandran, N., Hong, S. C., & Sime, M. J. (2018). Deep learning for automated screening of diabetic retinopathy: A review. Computers in Biology and Medicine, 101, 101-113.
2. Bellemo, V., Lim, Z. W., Lim, G., Nguyen, Q. D., Xie, Y., Yip, M. Y. T., ... & Wong, T. Y. (2018). Artificial intelligence using deep learning to screen for referable and vision-threatening diabetic retinopathy in Africa: A clinical validation study. The Lancet Digital Health, 1(1), e35-e44
3. Abràmoff, M. D., Lavin, P. T., Birch, M., Shah, N., & Folk, J. C. (2018). Pivotal trial of an autonomous AI-based diagnostic system for detection of diabetic retinopathy in primary care offices. npj Digital Medicine, 1(1), 1-8
4. Rajalakshmi, R., Subashini, R., Anjana, R. M., Mohan, V., Rani, P. K., & Deepa, M. (2018). Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence. Eye, 32(6), 1138-1144.