PROJECT DOCUMENTATION

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team Id:-PNT2022TMID48260

Submitted By:

M.Ummul Raihan - 912819106020 - Team Lead

M.P.Soniga - 912819106015

M.M.Sunmathi - 912819106017

M.Subhiksha - 912819106016

A.Jeyasri – 912819106005

Table of Contents:

DEEP LEARNING FUNDUS IMAGE ANALYSIS FOR EARLY DETECTION OF DIABETIC RETINOPATHY

ABSTRACT

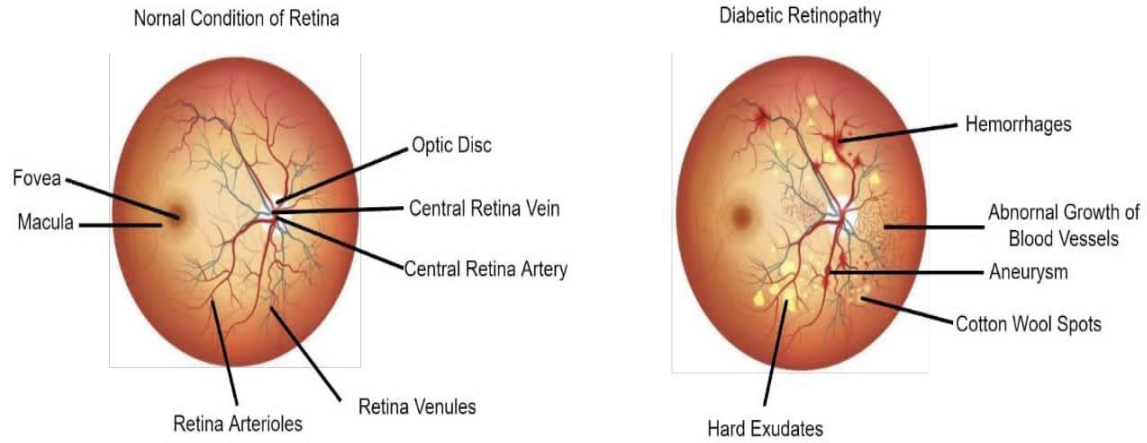
Diabetes problems can lead to a condition called diabetic retinopathy (DR), which permanently damages the blood vessels in the retina. If not treated, DR is a significant cause of blindness. The only DR treatments currently accessible are those that block or delay vision loss, which emphasizes the value of routine scanning with high-efficiency computer-based technologies to identify patients early. The major goal of this study is to employ a deep learning neural network to identify diabetic retinopathy in the retina's blood vessels. The NN classifier is put to the test using the input fundus image and DR database. It effectively contrasts retinal images and distinguishes between classes when there is a legitimate edge. For the resolution of the problems in the photographs, it is particularly useful. Here, it will be tested to see if the classification of diabetic retinopathy is normal or abnormal. Modifying the existing study's conclusion strategy, existing diabetic retinopathy techniques have sensitivity, specificity, and accuracy levels that are much lower than what is required for this research.

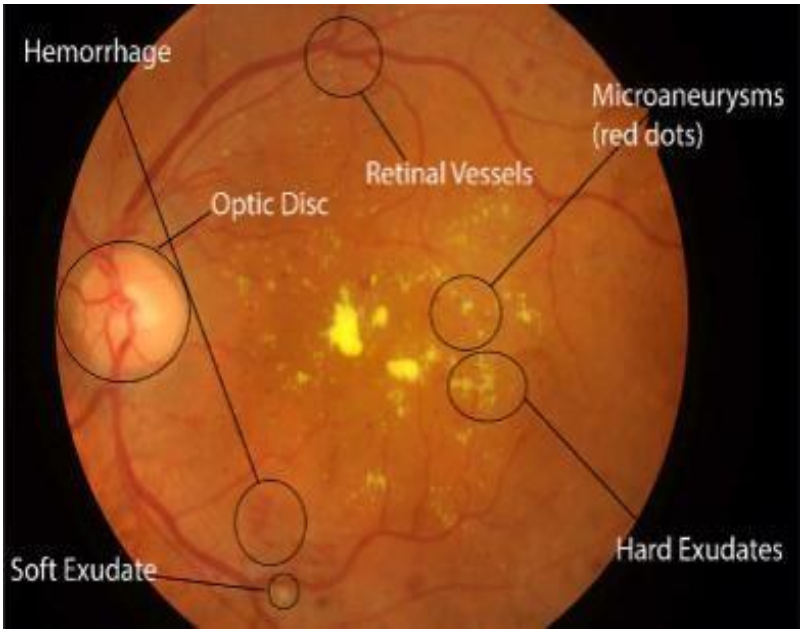
1. INTRODUCTION :

Diabetes is a determined organ disease that occurs when the pancreas of an individual is not able to release sufficient insulin. In a later stage, diabetes starts to impact the indirect structure of the body, even the retina. The most common diabetic individuals are infected by retinopathy. In the survey, we could see about 430 million diabetic individuals having the prospect of a visual disability. Diabetic retinopathy (DR) is a delicacy where the retina is hurt by virtue of sap openings in the retina. It happens only when diabetes hurts the infinitesimal deposit inside the retinal core. The sap in the retina structure will leak blood and liquid. The appearance of the integrant on the retina defines the classification, and the periods of diabetic retinopathy are perceived. A trained clinician is required to classify normal or not as it is a long cycle, and the clinician is also needed to check out and examine the dataset diversity of the fundus photographs. One of the fundamental difficulties is early identification, which is vital for treatment achievement. The specific ID of the diabetic retinopathy stage is interesting and requires master human comprehension of fundus pictures. Improvement of the recognition step is pivotal and can help a large number of individuals. Neural networks (NNs) have been successfully used to discover diabetic retinopathy in several contiguous patients.

Here, we proposed one more computation to recognize the retinal veins. The green channel will be chosen for picture investigation to precisely extricate vessels. The discrete wavelet change is utilized to upgrade the picture contrast for viable vessel identification. The directionality component of the multi-structure component strategy makes it a successful device in edge discovery. Hence, morphology administrators utilizing multi-structure components are applied to the upgraded picture to find the retinal picture edges. A short time later, morphological administrators by reproduction dispose of the edges not having a place with the vessel tree while attempting to protect the slight vessels unaltered. To extend the efficiency of the morphological heads by propagation, they were applied using multi-structure components.

Due to their capacity to identify and train the most discriminative features at the pixel level, neural networks, a subset of deep learning, have outperformed traditional machine learning-based techniques for medical imaging segmentation. In this study, we create a method for the automatic segmentation of retinal haemorrhages in the fundus picture using a neural network-based architecture. We use datasets with very high-quality photos tagged pixel by pixel to train neural networks. To teach the network to make judgments by delivering accurate information, the initial step is to search the data.





2. PROJECT DESIGN AND PLANNING:

2.1 IDEATION PHASE:

2.1.1 LITERATURE SURVEY

Approximately four hundred and twenty million people worldwide have been Diagnosed with diabetes mellitus. The prevalence of this disease has doubled in the past 30 years and is only expected to increase, particularly in Asia. Of those with diabetes approximately one-third are expected to be diagnosed with Diabetic retinopathy (DR), a chronic eye disease that can progress to irreversible vision loss. Early detection which is critical for good prognosis, release on skilled readers and is both labour and time-intensive. Automated techniques for Diabetic retinopathy diagnoses are essential to solving these problems.

1.Development and Validation of a Deep Learning Algorithm for Detection Of Diabetic Retinopathy in Retinal Fundus Photographs (2016)

- The reference standard used for this study was the majority decision of all Ophthalmologist graders.

- This means the algorithm may not perform as well for images with subtle findings that a majority of ophthalmologist would not identified.

- Another fundamental limitation arises from the nature of deep networks, In which the neural network was provided with only the image and Associated grade, without explicit definitions of features.

2. Development and Validation of a Deep Learning Algorithm for Detection Of Diabetic Retinopathy in Retinal Fundus Photographs (2018)

- The original study used non-public fundus images from EyePACS and Three hospitals in India for training. This Study used a different Eyepatch Data set from Kaggle the original study used the bench mark data set Messidor-2 to evaluate the algorithm’s performance. This study used the same data set. In the original study, ophthalmologist regraded all images for diabetic retinopathy, macular edema, and image gradeability.

- There was one diabetic retinopathy grade per image for data set, and assessed image gradability ourselves.

- The original study did not provide hyper-parameter settings. But some of these were later published.

3.Transfer Learning based Detection of Diabetic Retinopathy from small Dataset (2019)

- Transfer learning from an already trained deep convolutional network can be used to reduce the cost of training from scratch and to train with small training data for deep learning.

- In this work, they used a pretrained Inception-V3 model to take advantage of its Inception modules for Diabetic Retinopathy detection.

- In order to tackle the labelled data insufficiency problem, they subsampled a smaller version of the Kaggle Diabetic Retinopathy classification challenge dataset for model training, and tested the model's accuracy on a previously unseen data subset. Their technique could be used in other deep learning based medical image classification problems facing the labelled training data insufficiency**.**

4.Deep Learning Approach to Diabetic Retinopathy Detection (2020)

- One of the essential challenges is early detection, which is very important for treatment success.

- In this paper, they proposed an automatic deep-learning-based method for Stage detection of diabetic retinopathy by single photography of the Human fundus. Additionally, they propose the multistage approach to Transfer learning, which makes use of similar datasets with different Labelling.

- The presented method can be used as a screening method for early Detection of diabetic retinopathy with sensitivity and specificity of 0.99 And is ranked 54 of 2943 competing methods (quadratic weighted kappa Score of 0.925466) on APTOS 2019 Blindness Detection Dataset (13,000 Images).

REFERENCE

C. Sinthanayothin, J. F. Boyce, T. H. Williamson et al., "Automated detection of diabetic retinopathy on digital fundus images," Diabetic Medicine, vol. 19, no. 2, pp. 105–112, 2002.

U. R. Acharya, C. M. Lim, E. Y. K. Ng, C. Chee, and T. Tamura, "Computerbased detection of diabetes retinopathy stages using digital fundus images," Proceedings of the Institution of Mechanical Engineers – Part H: Journal of Engineering in Medicine, vol. 223, no. 5, pp. 545–553, 2009.

R. Afrin, Automatic detection and classification of diabetic lesions for grading diabetic retinopathy using fuzzy rule-based classification system, Khulna University of Engineering \& Technology (KUET), Bangladesh, pp. 10–12, 2019.

M. E. P. Martinez, A. D. Hughes, S. A. Thom, and K. H. Parker, "Improvement of a retinal blood vessel segmentation method using the insight segmentation and registration toolkit (ITK)," in Proceedings of the 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 892–895, August 2007.

M. Mendonc and A. Campilho, "Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction," IEEE Transactions on Medical Imaging, vol. 25, no. 9, pp. 1200–1213, 1213, Sep.

N. Yalin, S. Alver, and N. Uluhatun, "Classification of retinal images with deep learning for early detection of diabetic retinopathy disease," in Proceedings of the 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, May 2018.

P. R. Kshirsagar, P. P. Chippalkatti, and S. M. Karve, “Performance optimization of neural network using GA incorporated PSO,” Journal of Advanced Research in Dynamical and Control Systems, vol. 10, no. 4, 2018.

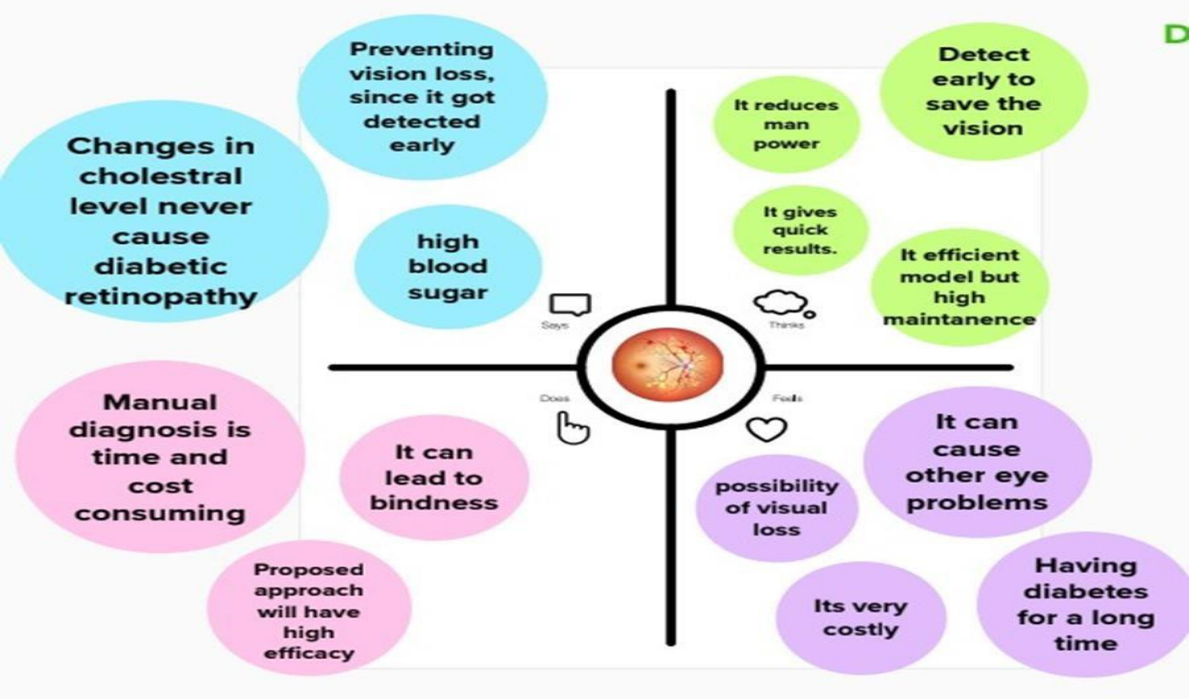
E. Kumar Kilari and S. Putta, “Delayed progression of diabetic cataractogenesis and retinopathy in STZ-induced diabetic rats,” Cutaneous and Ocular Toxicology, vol. 36, 2016.

T. Kauppi and H. Kälviäinen, “Simple and robust optic disc localization using colour decorrelated templates,” in Proceedings of the 10th Int. Conf. Advanced Concepts for Intell. Vision Syst, pp. 719–727, Springer-Verlag, Berlin, Germany, October 2008.

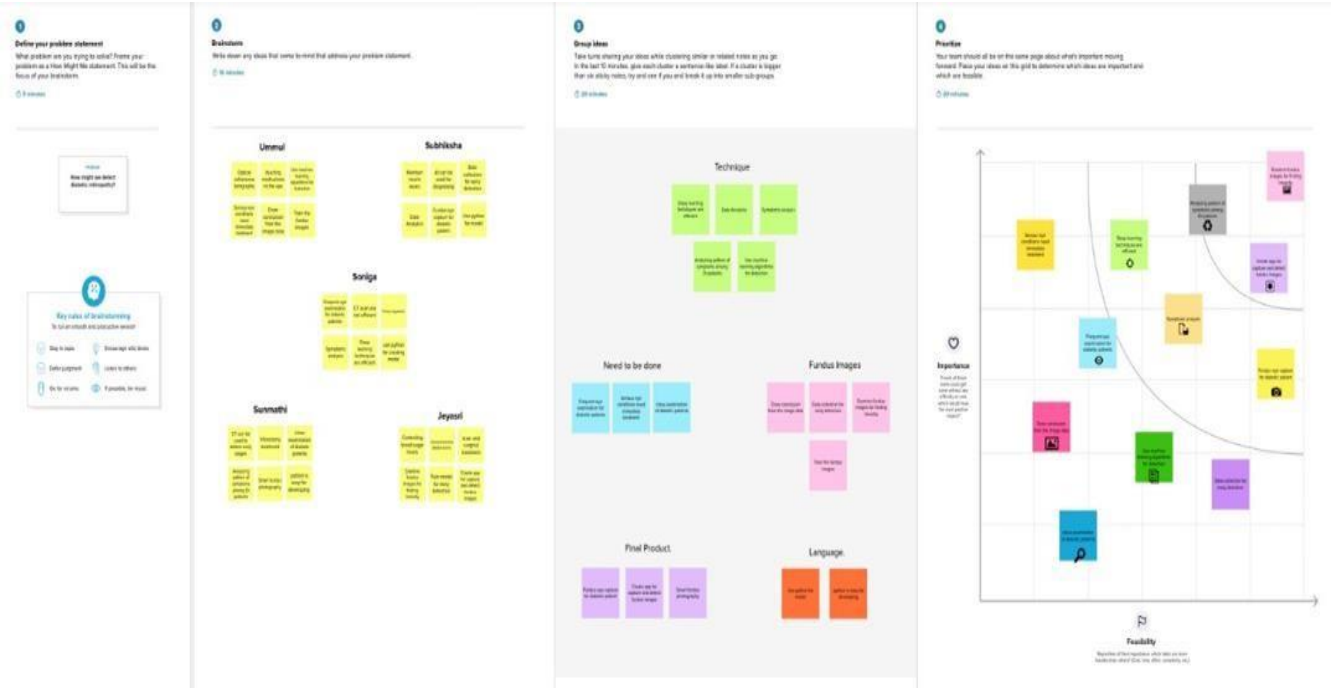
S. Mirajkar and M. M. Pati, “Feature extraction of diabetic retinopathy images,” International Journal of Computer Application, 2013.

2.1.2 Empathy Map Canvas :

Diabetic Retinopathy (DR) is a Common complication of Diabetes mellitus, which causes Lesions on the retina that affect Vision.If it is not detected early, it Can lead to blindness. Unfortunately, DR is not a Reversible process, and Treatment only sustains vision. DR early detection and Treatment can signifcantly Reduce the risk of vision loss. The manual diagnosis process of DR retina fundus images by Ophthalmologists is time, effort And cost-consuming and prone To misdiagnosis unlike Computer-aided diagnosis Systems.



2.1.3 IDEATION AND BRAINSTORMING:-



2.1.4 PROBLEM STATEMENT :-

Diabetic Retinopathy (DR) is a common complication of diabetes Mellitus, which causes lesions on the retina that affect vision. If it is not Detected early, it can lead to blindness.

1.Who does the problem affect?

Persons who have Diabetes mellitus

2.What are the boundaries of the Problem?

The longer you have diabetes and The less controlled your blood sugar Is, the more likely you are to Develop this eye complication.

3.What is the issue?

• Spots or dark strings Floating in your vision (floaters)

• Blurred vision

• Fluctuating vision

• Dark or empty areas In your vision

• Vision loss

4.When does the issue occur?

Developing diabetes when Pregnant (gestational diabetes) or Having diabetes before becoming Pregnant can increase risk of Diabetic retinopathy.

5.Where does the issue occur?

The manual diagnosis process of DR retina fundus images by Ophthalmologists is time, effort and Cost-consuming and prone to Misdiagnosis unlike computer-aided Diagnosis systems.

6.Why is it important that we fix the Problem?

If it is not detected early, it can lead To blindness.Unfortunately, diabetic Retinopathy is not a reversible Process, and treatment oSustains vision.

7.What solution to solve this issue?

DR early detection and treatment Can significantly reduce the risk of Vision loss

2.2. PROJECT DESIGN PHASE 1

2.2.1 PROPOSED SOLUTION

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).At first, diabetic retinopathy might cause no symptoms or only mild vision problems. But it can lead to blindness. The condition can develop in anyone who has type 1 or type 2 diabetes. The longer you have diabetes and the less controlled your blood sugar is, the more likely you are to develop this eye complication
2.	Idea / Solution description	Diabetic retinopathy is not a reversible process,and treatment only sustains vision. DR early detection and treatment can significantly reduce the risk of vision loss. The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis systems. So, deep learning techniques can be used for early detection of diabetic retinopathy that can prevent blindness and other eye related diseases.
3.	Novelty / Uniqueness	This model provides the patient with the result whether they have a serious condition or normal condition. The prediction comes with different levels of illness and helps to diagnose properly.

4.	Social Impact / Customer Satisfaction	Since, Diabetic retinopathy is irreversible, early detection helps many people from losing eyesight and other complicated diseases. The manual screening costs more than this model hence it is more feasible for customers that they can take this screening without any hardships
5.	Business Model (Revenue Model)	We can collaborate with the health care centers and diabetic diagnosis centers for regular screening of diabetic retinopathy whenever the diabetic patient comes to check their diabetic level. We can create awareness among people because many people have no idea about the effects diabetic retinopathy, it may result in many screening tests in future.
6.	Scalability of the Solution	The solution with the transfer learning model offers a better solution for diabetic retinopathy and can be detected at an early stage. The model developed using deep learning technology can be implemented on many clinical examinations. This system is versatile as it can learn from any datasets. It gives higher performance than manual examination.

2.2.2. PROBLEM SOLUTION FIT

Project Title: Deep Learning Fundus Image Analysis For Early Detection of Diabetic retinopathy

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID48260

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

The early detection important for the diabetic patients as diabetic retinopathy is irreversible. The Diabetic retinopathy can be detected using the fundus image of the patient and can be stored in the database.This is more useful than the manual examination

6. CUSTOMER CONSTRAINTS CC

The diabetic retinopathy does not nave any specific symptoms so they fail to notice the illness. Many people do not know about diabetic retinopathy and its adverse reaction.

5. AVAILABLE SOLUTIONS AS

Laser treatment to treat the growth of new blood vessels at the back of the eye (retina) in cases of proliferative diabetic retinopathy, and to stabilize some cases of maculopathy. eye injections - to treat severe maculopathy that's threatening your sight.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

The problem is once the diabetic retinopathy is severe, it cannot be done.

And the severity of diabetic retinopathy results in serious eye illness and also results in losing vision. So, the early detection is important if the patient has diabetes.

9. PROBLEM ROOT CAUSE RC

Diabetic retinopathy is caused by changes in the blood vessels of the retina, the light-sensitive layer of tissue at the back of the inner eye. In some people with diabetic retinopathy, the blood vessels in the retina may swell and leak fluid. In others, abnormal new blood vessels grow on the surface of the retina.

7. BEHAVIOUR BE

This model helps in the early detection of diabetic retinopathy using the fundus images.
It consumes less time than the manual examination.
Also, accuracy is more compared to other techniques.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

The triggers in diabetic retinopathy patients are
- Spots or dark strings floating in your vision (floaters)
- Blurred vision.
- Fluctuating vision.
- Dark or empty areas in vision.
- Vision loss.

4. EMOTIONS: BEFORE / AFTER EM

Before: Adverse emotional responses include fear, anxiety, vulnerability, guilt, loss of confidence, anger, stress and self-perception issues.
After: Early detection and diagnosis gives sense of hope among patients

10. YOUR SOLUTION SL

Our solution involves the deep learning model with fundus images that detect the severity of the diabetic retinopathy among diabetic patients and the apt diagnosis done after the early detection.

8.CHANNELS OF BEHAVIOUR CH

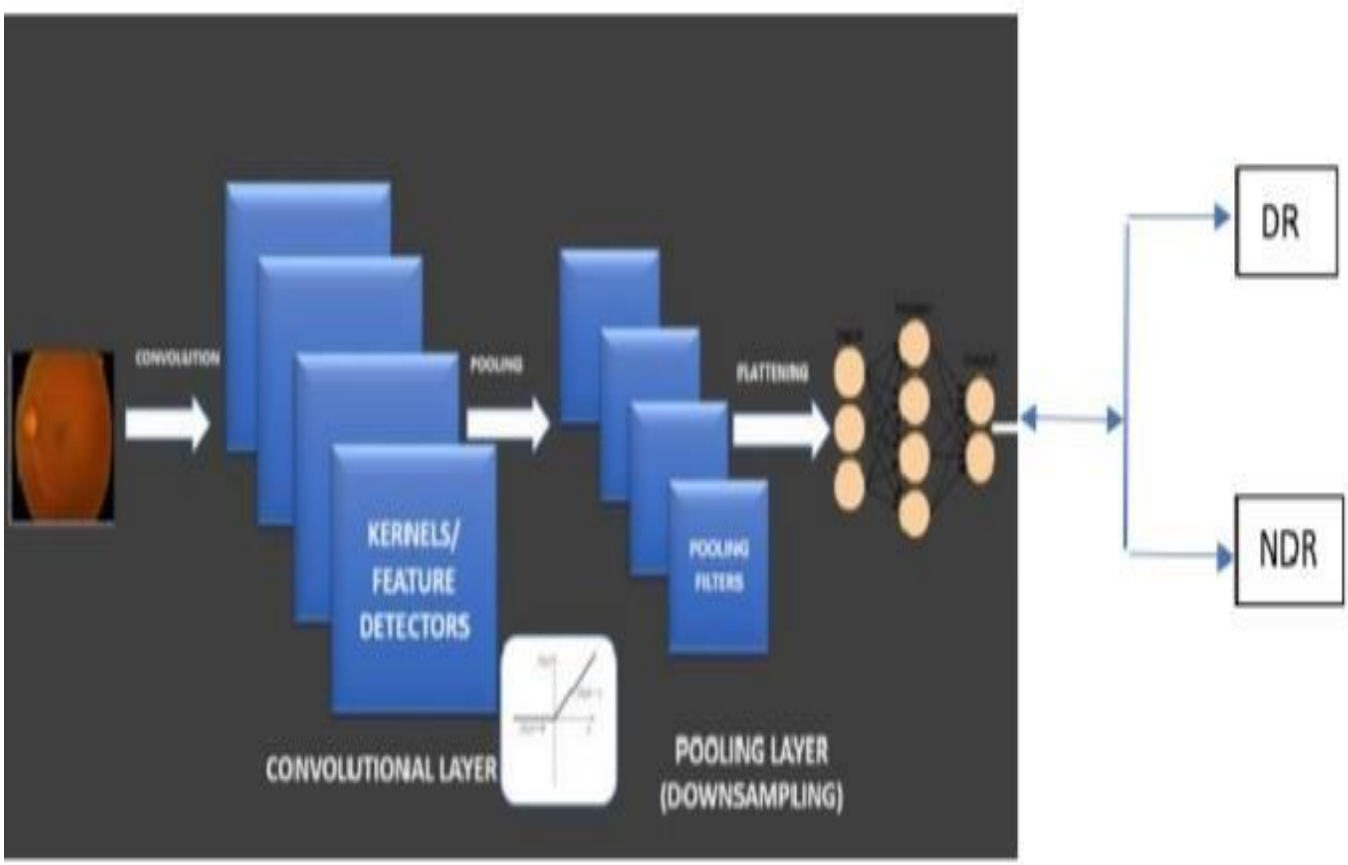
The diabetic patients have to take the eye examination in the regular interval time. Then only retinopathy can be detected early and proper diagnosis can be done.

2.2.3 Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders. • Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram



2.3 PROJECT DESIGN PHASE 2

2.3.1 SOLUTION REQUIREMENT:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User registration	Registration through phone number Registration through Gmail
FR-2	User Confirmation	Confirmation via OTP Confirmation via Mail
FR-3	Describe what the product does	Our project can detect early changes in your retina before you notice any difference in your eyesight
FR-4	Focus on user requirements	Reduce the risk of visual loss and blindness in patients with retinal complications of diabetes.
FR-5	Usually defined by user	Fundus image obtained from patients.
FR-6	Define product features	To an advanced eye screening technology by which eye related diseases can be detected at an early stage.

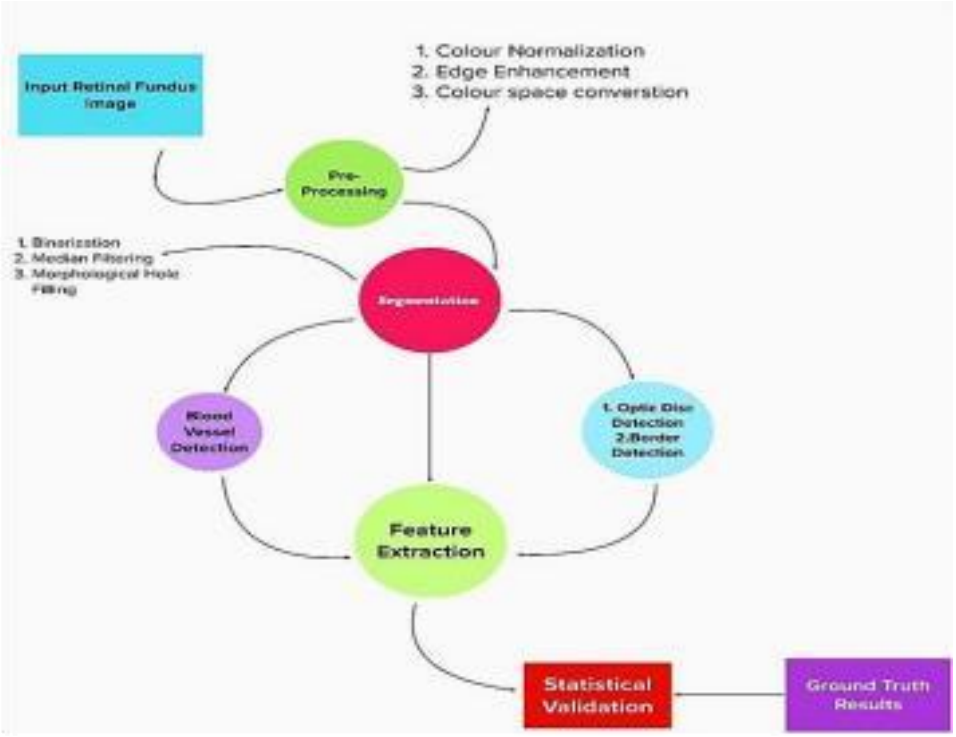
Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR NO:	Non-Functional Requirement	Description
NFR-1	**Usability**	Assuring that a software can effectively perform one or more defined functions.
NFR-2	**Security**	Permission granted only by the administrator of the system.
NFR-3	**Reliability**	If the system update fails or bugs in the code even though the system can roll back to its initial state.
NFR-5	**Availability**	Health care affordability, quality and accessibility is made easier using the device.
NFR-6	**Scalability**	The product must hold stable even when multiple users are using it at the same times.

2.3.2 DATA FLOW DIAGRAM

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



The retinal fundus images are commonly used for detection and analysis of diabetic retinopathy disease in clinics.

- Pre-processing of raw retinal fundus images are performed using extraction of green channel, histogram equalization, image enhancement and resizing techniques.
- The segmentation of retinal vasculature from eye fundus images is a fundamental task in retinal image analysis.
- The computer aided automatic detection and segmentation of blood vessels through the elimination of optic disc (OD) region in retina.
- The retinal blood vessels are detected using mathematical binary morphological operations.

- Feature extraction from the fundus images for the diagnosis of Diabetic Retinopathy refers to an advanced eye screening technology by which eye related diseases can be detected at an early stage.

2.3.3 TECHNOLOGY ARCHITECTURE:-

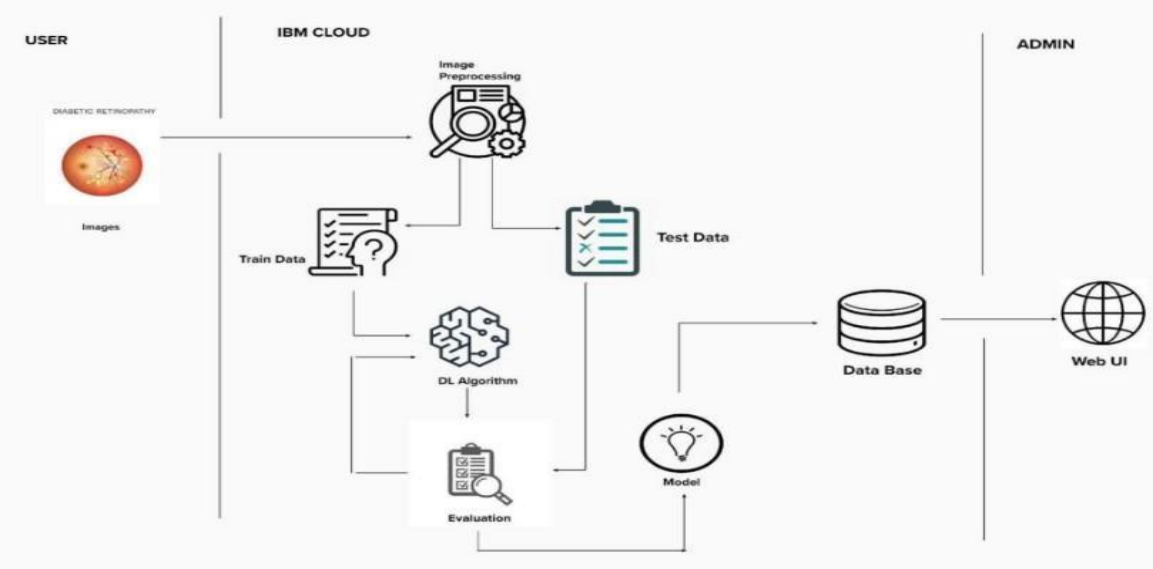


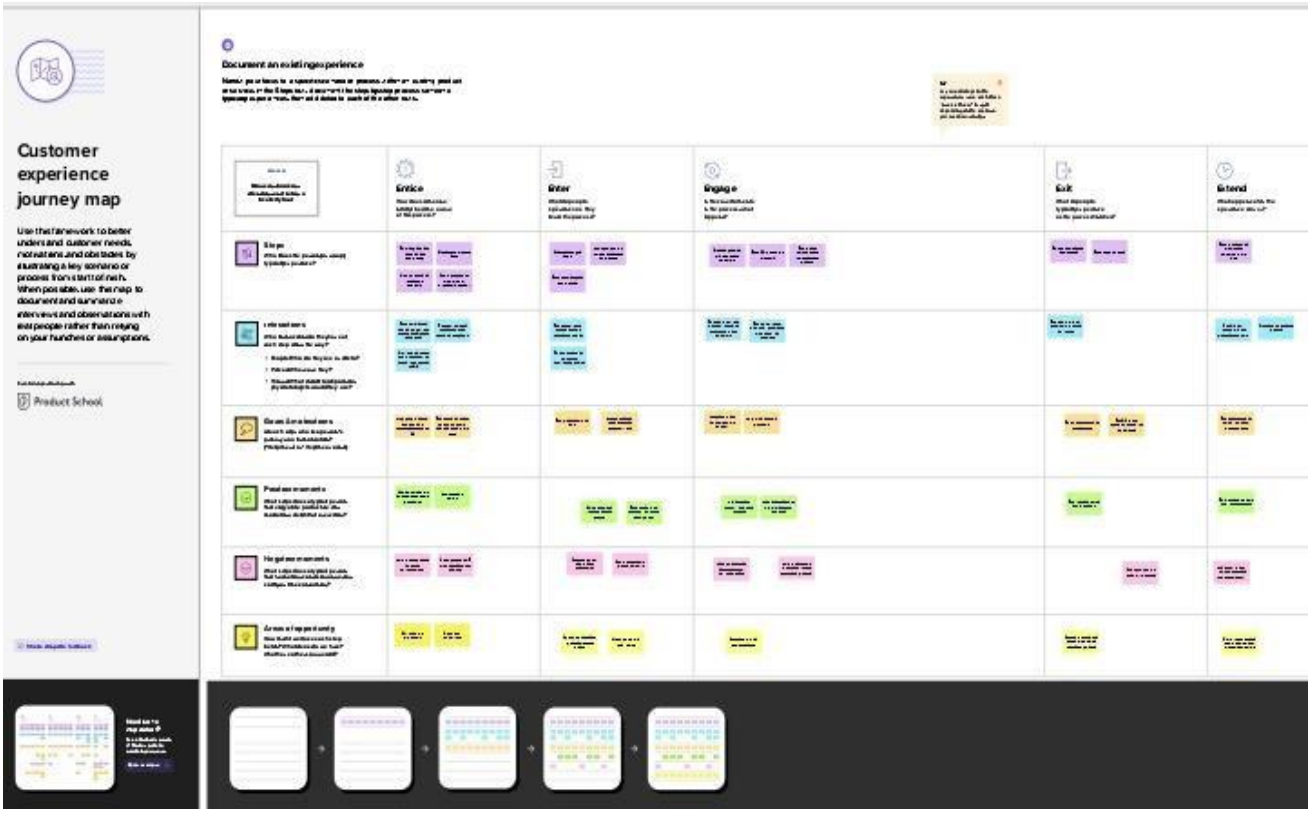
Table-1:Components & Technologies:

1	User Interface	Web UI	HTML, CSS, JavaScript, Python
2	Application logic-1	Image Preprocessing	Keras,Tensorflow,Numpy
3	Application logic-2	CNN Model	Keras,Tensorflow,Numpy
4	Application logic-3	Web UI Application	Flask
5	Database	DR Images (Jpeg,Png,Jpg,Etc.,)	Uploads Folder
6	File storage	File Storage Requirements (Only If Necessary)	IBM Block Storage, GoogleDrive
7	External Api	Keras	Image Processing API
8	Deep Learning Model	Inception V3 Architecture	Pre-Trained Convolution NeuralNetwork Model
9	Infrastructure (Server)	Application Deployment on Webserver	Flask-A PythonWSGI HTTP Server.

Table-2:Application characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Flask	Flask Frameworks
2.	Security Implementations	CSRF Protection,Secure Flag For Cookies	Flask-WTF, Session Cookie Secure
3.	Scalable Architecture	Micro-Services	Micro Web Application FrameworkBy Flask

2.3.4 CUSTOMER JOURNEY MAP



2.4. PROJECT PLANNING

2.4.1 SPRINT DELIVERY PLAN

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Registration	USN-1	As a user, I can register through website either by email id or phone number with password.	5	High	Ummul Raihan.M Sunmathi. M. M
Sprint-3	Login	USN-2	As a user, I can login to the site by the given Login credentials.	5	High	Subhiksha. M Sunmathi. M. M
Sprint-3	Upload image	USN-3	As a user,I can upload my data in the form of pdf, doc etc.	2	Medium	Subhiksha. M Jeya Sri . A
Sprint-3	Admin login	USN-4	As an admin I can login to the site and analyze the user data.	2	High	Soniga. M. P Jeya Sri . A

Sprint-1	Data collection	USN-5	As an admin, I can collect the dataset related to the DR from source.	5	Low	Soniga. M. P Ummul Raihan.M
Sprint-1	Create model	USN-6	As an admin,I can create the model for prediction.	5	High	Subhiksha. M Jeya Sri. A
Sprint-2	Test the model	USN-7	As an admin, I can test the model for prediction.	6	High	Jeya Sri . A Sunmathi. M. M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Diagnosis	USN-8	As a user I can get the diagnosis result on the application and follow up with treatments.	6	High	Soniga. M. P Sunmathi. M. M
Sprint-4	Train the model	USN-9	As an admin, I can train the model for prediction.	10	High	Subhiksha. M jeyasri.A

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	12	05 Nov 2022
Sprint-3	14	6 Days	07 Nov 2022	12 Nov 2022	14	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

AV1=Sprint duration/velocity=10/6=1.66

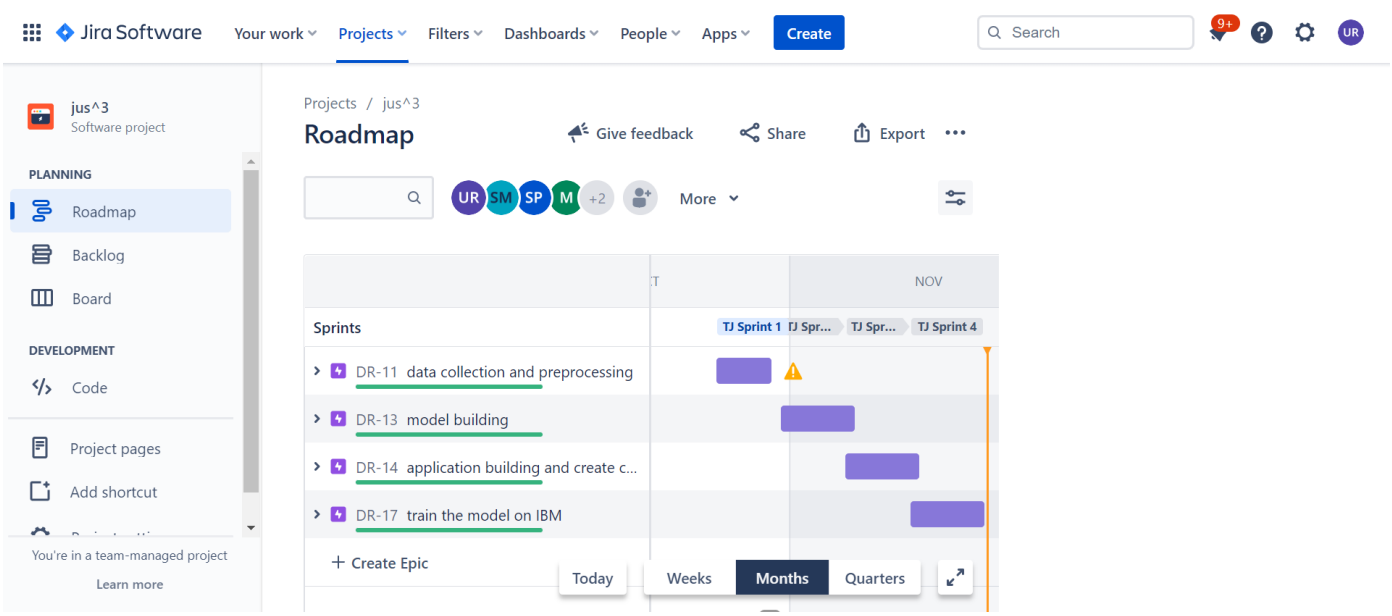
AV2=Sprint duration/velocity=12/6=2

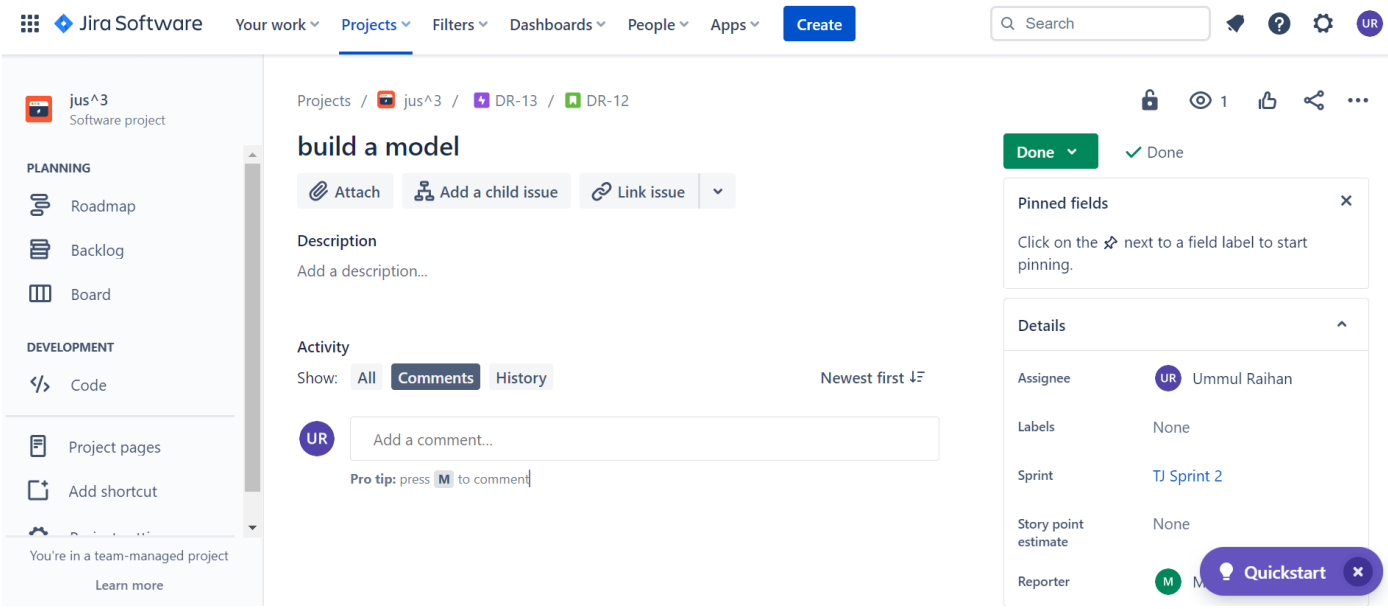
AV3=Sprint duration/velocity=14/6=2.3

AV4=Sprint duration/velocity=10/6=1.66

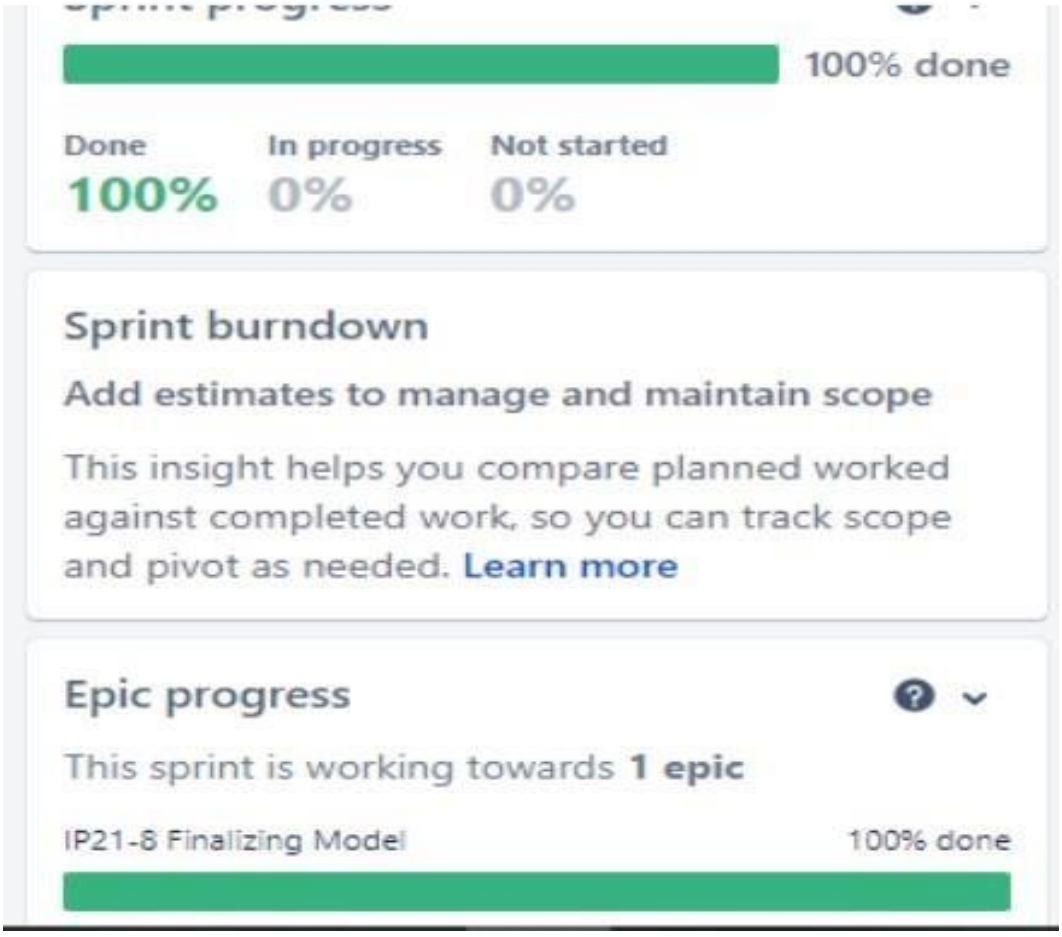
Burndown chart:

2.4.2 JIRA SCREENSHOTS:




Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
jus^3
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
You're in a team-managed project
Learn more
Projects / jus^3 / DR-13 / DR-12
build a model
Attach
Add a child issue
Link issue
Description
Add a description...
Activity
Show: All Comments History
Newest first
Add a comment...
Pro tip: press M to comment
Done
Done
Pinned fields
Click on the next to a field label to start pinning.
Details
Assignee
Ummul Raihan
Labels
None
Sprint
TJ Sprint 2
Story point estimate
None
Reporter
Quickstart

Insights:


100% done
Done
100%
In progress
0%
Not started
0%
Sprint burndown
Add estimates to manage and maintain scope
This insight helps you compare planned worked against completed work, so you can track scope and pivot as needed. Learn more
Epic progress
This sprint is working towards 1 epic
IP21-8 Finalizing Model
100% done

2.4.3 MILESTONE AND ACTIVITY LIST

TITLE	DESCRIPTION	DATE
Literature survey & information gathering	It is to understand and gathering information of the existing solutions, technical papers and research publications to the project.	29 AUGUST 2022
Prepare empathy map	It generally captures the user pains and gains of the project.	30 AUGUST 2022
Ideation	It is a process of generating and developing new creative ideas for the project.	02 SEPTEMBER 2022
Proposed solution	After the process of analyzing the problems the required practical solutions are proposed and rectifying the issues.	19 SEPTEMBER 2022
Problem solution fit	It generally means that the problem found with our customer and the solution is realized and concludes their problem..	22 SEPTEMBER 2022
Solution architecture	It is an architectural description for a solution with the technical applications and operations used.	23 SEPTEMBER 2022
Customer journey	It is to understand better how the user experiences our product	03 OCTOBER 2022
Data Flow diagram	It is a visual representation of the information flows within a system and how data enters and leaves the system.	05 OCTOBER 2022
Functional and non-functional requirements	Functional requirements describe the system behavior under the specific conditions. Nonfunctional requirements define how the system should perform.	07 OCTOBER 2022
Technology architecture	The architecture specifies the hardware,software,access methods and protocols used throughout the system.	08 OCTOBER 2022
Prepare milestone and activity list	It describes the start date and end date of the project phase	18 OCTOBER 2022
Project development-Delivery of sprint-1,2,3&4	It is a process of planning and allocating resources to fully develop a project.	IN PROGRESS

3. PROJECT DEVELOPMENT PHASE

Feature 1:

We have devloped a website which authenticates users and help them upload and check the seriousness of the diabetics.

Feature 2:

We have devloped a multilayer deep convolutional nueral network that classifies the user image of a eye to which extense has the disease diabetics has been affected.The model will classify the images into 5 categories of diabetics and report them on asking for prediction. We have also devloped a messaging service for recieiving message for the type of diabetics.

SPRINT-1

3.1 DATA COLLECTION

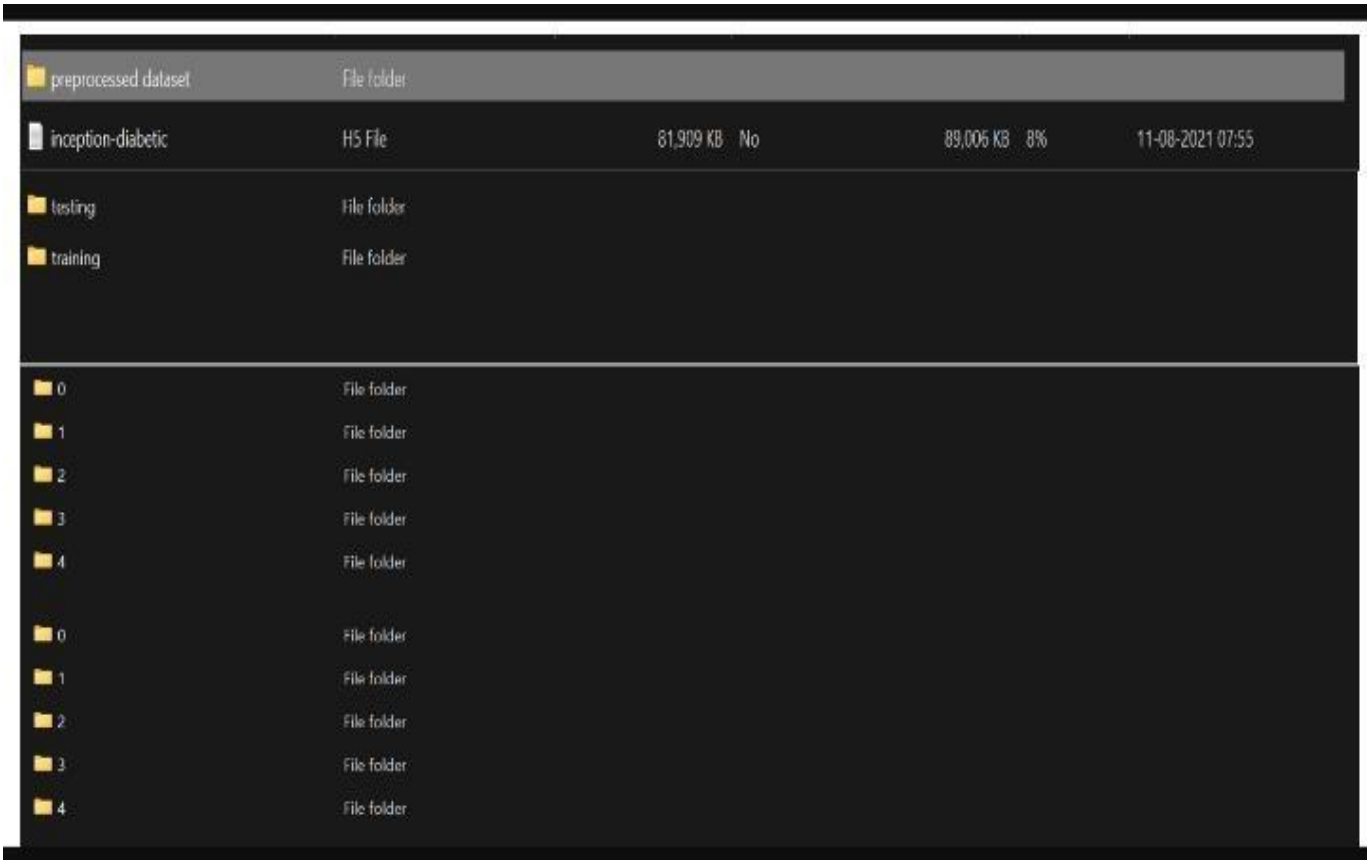
3..1.1 CREATE TESTING AND TRAINING PATH

```
imageSize=[299,299]
```

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

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

3.1.2 DOWNLOAD DATASET



3.2 DATA PREPROCESSING

3.2.1 Pre-Processing/Apply ImageDataGenerator Class

```
x_train=train_datagen.flow_from_directory(r"preprocessed dataset\preprocessed dataset\training",target_size=(299,299),batch_size=
```

```
Found 3662 images belonging to 5 classes.
```

```
x_test = test_datagen.flow_from_directory('preprocessed dataset/preprocessed dataset/testing', target_size = ( 299 , 299 ) ,batch
```

```
Found 734 images belonging to 5 classes.
```

3.2.2 Data Pre-Processing/Configure ImageDataGenerator class

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

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

3.2.3 Data Pre-Processing/Import Libraries

```
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
```

3.2.4 Sprint 1/Sprint_1

127 lines (127 sloc) | 2.83 KB ...

In [2]:
```
from tensorflow.keras.layers imp
from tensorflow.keras.models imp
from tensorflow.keras.preprocess
from tensorflow.keras.preprocess
from tensorflow.keras.applicatic
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
```

In [18]:
```
imageSize=[299,299]
```

In [8]:
```
trainPath=r"/content/preprocesse
```

In [9]:
```
testPath=r"/content/preprocessed
```

In [10]:
```
train_datagen=ImageDataGenerator
```

In [11]:
```
test_datagen=ImageDataGenerator(
```

In [17]:
```
x_train=train_datagen.flow_from_
```

```
Found 3662 images belonging to 5
classes.
```

In [18]:
```
x_test = test_datagen.flow_from_
```

```
Found 734 images belonging to 5
classes.
```

In [ ]:
```

```

SPRINT -2 3.3 MODEL BUILDING

3.3.1 Model Building/Adding Dense Layers

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

```
model = Model ( inputs = xception.input , outputs = prediction )
```

```
model.summary()
```

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         [(None, 299, 299, 3)      0
_________________________________________________________________________________________________
block1_conv1 (Conv2D)        (None, 149, 149, 32)      864         input_1[0][0]
_________________________________________________________________________________________________
block1_conv1_bn (BatchNormaliza (None, 149, 149, 32)   128         block1_conv1[0][0]
_________________________________________________________________________________________________
block1_conv1_act (Activation)  (None, 149, 149, 32)    0           block1_conv1_bn[0][0]
_________________________________________________________________________________________________
block1_conv2 (Conv2D)        (None, 147, 147, 64)      18432       block1_conv1_act[0][0]
_________________________________________________________________________________________________
block1_conv2_bn (BatchNormaliza (None, 147, 147, 64)   256         block1_conv2[0][0]
_________________________________________________________________________________________________
block1_conv2_act (Activation)  (None, 147, 147, 64)    0           block1_conv2_bn[0][0]
_________________________________________________________________________________________________
block2_sepconv1 (SeparableConv2 (None, 147, 147, 128 8768          block1_conv2_act[0][0]
```

3.3.2 Sprint 2/Model Building/Configure Learning process

```
model.compile (
 loss = 'categorical_crossentropy' ,
 optimizer = 'adam',
 metrics = [ 'accuracy' ])
```

3.3.3 Sprint 2/Model Building/Pre-trained CNN model

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

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [==============================] - 42s 1us/step
83697664/83683744 [==============================] - 42s 1us/step
```

```
]: for layer in xception.layers :
    layer.trainable = False
```

```
]: x = Flatten ( ) ( xception.output )
```

3.3.4 Sprint 2/Model Building/Save the model

```
model.save ( "Updated-xception-diabetic-retinopathy.h5")
```

3.3.5 Sprint 2/Model Building/Train the model

```
r = model.fit(
 x_train ,
 validation_data = x_test ,
 epochs = 50 ,
 steps_per_epoch = len ( x_train ) // 32 ,
 validation_steps = len ( x_test ) // 32)
```

```
Epoch 1/50
3/3 [==============================] - 35s 11s/step - loss: 10.4952 - accuracy: 0.4167
Epoch 2/50
3/3 [==============================] - 31s 9s/step - loss: 13.1979 - accuracy: 0.5729
Epoch 3/50
3/3 [==============================] - 29s 9s/step - loss: 11.9821 - accuracy: 0.5104
Epoch 4/50
3/3 [==============================] - 28s 8s/step - loss: 9.1480 - accuracy: 0.5000
Epoch 5/50
3/3 [==============================] - 29s 9s/step - loss: 7.1723 - accuracy: 0.5104
Epoch 6/50
3/3 [==============================] - 29s 9s/step - loss: 5.8842 - accuracy: 0.5000
Epoch 7/50
3/3 [==============================] - 30s 9s/step - loss: 7.8335 - accuracy: 0.5938
Epoch 8/50
3/3 [==============================] - 34s 10s/step - loss: 4.6266 - accuracy: 0.6562
Epoch 9/50
3/3 [==============================] - 29s 9s/step - loss: 2.6728 - accuracy: 0.7812
Epoch 10/50
3/3 [==============================] - 33s 10s/step - loss: 3.7426 - accuracy: 0.6146
Epoch 11/50
```

SPRINT -3 3.4 Sprint 3/Application Building

3.4.1 Sprint 3/Application Building/Build Python Code

```
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 # Authenticate using an IAM
API key client = Cloudant.iam('7a432718-925c-4d97-b679-
d49131ffae12bluemix','aQskDQNm0R_p8myc2CcIXTJ3COwVpWeN3V
KVrddm4MjP'
, connect=True)
# Create a database using an initialized client my_database =
client.create_database('my_database') model =
load_model(r"Updated-Xception-diabetic-retinopathy.h5") app =
Flask(__name__) # default home page or route
@app.route('/') def
index():
    return render_template('index.html')
@app.route('/index.html') def
home():
    return render_template("index.html")
# registration page
@app.route('/register')
def register():
    return render_template('register.html')
@app.route('/afterreg', methods=['POST'])
def afterreg():
```

```
    x = [x for x in request.form.values()]
print(x)     data = {
    '_id': x[1], # Setting _id is optional
    'name': x[0],
    'psw':x[2]
    }     print(data)        query = {'_id': {'$eq':
data['_id']}}         docs =
my_database.get_query_result(query)
print(docs)        print(len(docs.all()))
if(len(docs.all())==0):
        url = my_database.create_document(data)
        #response = requests.get(url)         return
render_template('register.html', pred="Registration Successful, please
login using your details")     else:
        return render_template('register.html', pred="You are already a
member, please login using your details")
# login page
@app.route('/login') def
login():
    return render_template('login.html')
@app.route('/afterlogin',methods=['POST']) def
afterlogin():
    user = request.form['_id']     passw =
request.form['psw']     print(user,passw)
query = {'_id': {'$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')
@app.route('/logout')     def
logout():
   return render_template('logout.html')
@app.route('/prediction') def
prediction():
   return render_template('prediction.html')
@app.route('/result',methods=["GET","POST"]) def res():     if request.method=="POST":         f=request.files['image']
basepath=os.path.dirname(__file__) #getting the current path i.e where app.py is present
        #print("current path",basepath)
filepath=os.path.join(basepath,'uploads',f.filename) #from anywhere in the system we can give image but we want that image later  to process so we are saving it to uploads folder for reusing         #print("upload folder is",filepath)         f.save(filepath)
img=image.load_img(filepath,target_size=(299,299))
x=image.img_to_array(img)#img to array
x=np.expand_dims(x,axis=0)#used for adding one more dimension
        #print(x)         img_data=preprocess_input(x)
prediction=np.argmax(model.predict(img_data), axis=1)
```

```
    #prediction=model.predict(x)#instead of predict_classes(x) we can use
predict(X) ---->predict_classes(x) gave error          #print("prediction is
",prediction)
        index=['No Diabetic Retinopathy', 'Mild DR', 'Moderate DR', 'Severe
DR', 'Proliferative DR']
        #result  =  str(index[output[0]])          result=str(index[
prediction[0]])          print(result)          return
render_template('prediction.html',prediction=result)
        """ Running our application """
if __name__ == "__main__":
app.run()
```

3.4.2 Sprint 3/Application Building/Building Html Pages

3.4.2.1 Sprint 3/Application Building/Building Html Pages/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initialscale=1.0">


    <!--Bootstrap -->
    <link                                                rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.
css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgF
```

```
AW/dAiS6JXm" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpG
FF93hXpG5KkN" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.m
in.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskv
Xusvfa0b4Q" crossorigin="anonymous"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+
76PVCmYl" crossorigin="anonymous"></script>



    <script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>
    <link
href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto
&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="../static/style.css">
    <!-- <script defer src="../static/js/main.js"></script> -->
    <title>Diabetic Retinopathy Classification</title>
</head>
<body>
    <header id="head" class="header">
   <section  id="navbar">
          <h1 class="nav-heading"></i>Diabetic Retinopathy
Classification</h1>
```

```
      <div class="nav--items">
         <ul>
                              <li><a  href="{{
url_for('index')}}">Home</a></li>
                              <li><a href="{{
url_for('login')}}">Login</a></li>
                              <li><a href="{{
url_for('register')}}">Register</a></li>
            <!-- <li><a href="#about">About</a></li>
            <li><a href="#services">Services</a></li> -->
            <li><a href="{{ url_for('login')}}">Prediction</a></li>
         </ul>
      </div>
   </section>
   <section id="slider">
    <div id="carouselExampleIndicators" class="carousel"
dataride="carousel">
       <ol class="carousel-indicators ">
          <li data-target="#carouselExampleIndicators" data-slide-to="0"
class="active "></li>
          <li data-target="#carouselExampleIndicators" data-
slideto="1"></li>
          <li data-target="#carouselExampleIndicators" data-
slideto="2"></li>
          <li data-target="#carouselExampleIndicators" data-
slideto="3"></li>
          <li data-target="#carouselExampleIndicators" data-
slideto="4"></li>
       </ol>
```

```
    <div class="carousel-inner">


        <div class="carousel-item active">
            <img class="d-block w-100" src="../static/img/first.png"
alt="First slide">
        </div>
        <div class="carousel-item">
            <img class="d-block w-100" src="../static/img/second.png"
alt="Second slide">
        </div>
        <div class="carousel-item">
            <img class="d-block w-100" src="../static/img/third.png"
alt="Third slide">
        </div>
    </div>
    <a class="carousel-control-prev" href="#carouselExampleIndicators"
role="button" data-slide="prev">
        <span class="carousel-control-prev-icon"
ariahidden="true"></span>
        <span class="sr-only">Previous</span>
    </a>
    <a class="carousel-control-next" href="#carouselExampleIndicators"
role="button" data-slide="next">
        <span class="carousel-control-next-icon"
ariahidden="true"></span>
        <span class="sr-only">Next</span>
    </a>
  </div>
```

```
  </section>
</header>
<section id="about">
   <div class="top">
      <h3 class="title text-muted">
         ABOUT PROJECT
      </h3>
      <div class="line"></div>
   </div>
<div class="body">
<div class="left">
   <h2>Problem:</h2>
   <p>
      Diabetic Retinopathy (DR) is a common complication of diabetes mellitus, which causes lesions on the retina that effect vision. If it is not detected early, it can lead to blindness. Unfortunately, DR is not a reversible process, and treatment only sustains vision. DR early detection and treatment can significantly reduce the risk of vision loss. The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis systems.

               </p>
</div>
<div class="right">
   <h2>Solution:</h2>
   <p>
      In this project, we will be building a Transfer learning model that can detect and classify types of Diabetic Retinopathy. A web application is
```

integrated with the model, from where the user can upload a Diabetic Retinopathy (DR) image like Mild DR, Severe DR, etc., and see the analyzed results on UserInterface.

</p>

</section>

<section id="services">

<h3 class="title text-muted">WE CLASSIFY</h3>

<img src="../static/img/testimonials/NoDR.png" class="card-imgtop" alt="cardboard">

<h5 class="card-title text-muted">No DR</h5>

<img src="../static/img/testimonials/MildDR.png" class="card-imgtop" alt="bottle">

<h5 class="card-title text-muted">Mild DR</h5>

<img src="../static/img/testimonials/ModerateDR.png" class="cardimg-top" alt="metal">

```
      <div class="card-body text-muted">
            <h5 class="card-title text-muted">Moderate DR</h5>
</div>
    </div>
  <div class="card" style="width: 25rem;">
      <img src="../static/img/testimonials/SevereDR.png" class="card-
imgtop" alt="page">
      <div class="card-body text-muted">
                  <h5 class="card-title text-muted">Severe DR</h5>
 </div>
    </div>
  <div class="card" style="width: 25rem;">
      <img src="../static/img/testimonials/ProliferativeDR.png"
class="card-img-top" alt="plastic">       <div class="card-body
text-muted">
                  <h5 class="card-title text-muted">Proliferative
DR</h5>
 </div>
    </div>
  </div>
</section>
<section id="footer">
  <p>Copyright © 2021. All Rights Reserved</p>
  <div class="social">
    <a href="#" target="_blank"><i class="fab fa-2x fa-
twittersquare"></i></a>
     <a href="#" target="_blank">
      <i class="fab fa-2x  fa-linkedin"></i></a>
```

```
        <a href="#">
            <i class="#"></i>
        </a>
    </div>
</section>
</body>
</html>
```

Sprint 3/Application Building/Building Html Pages/login.html

3.4.2.2 Sprint 3/Application Building/Building Html Pages/login.html

ml >

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title> Login Page</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>
<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:30
0' rel='stylesheet' type='text/css'>
<!link rel="stylesheet" href="{{ url_for('static', filename='css/style.css')
}}">
```

```
<link href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans'
rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'>



<style>
.header {
                    top:0;
margin:0px;                    left: 0px;
right: 0px;                position: fixed;
background-color: #28272c;                color:
white;                    box-shadow: 0px 8px
4px grey;                 overflow: hidden;
      padding-left:20px;                  font-
family: 'Josefin Sans';                font-size:
2vw;                      width: 100%;
height:8%;
                    text-align: center;
              }
.topnav {
  overflow: hidden;   background-
color: #333;
}
```

```
.topnav-right a {  float: left;
color: #f2f2f2;  text-align:
center;  padding: 14px 16px;
text-decoration: none;  font-
size: 18px; } .topnav-right
a:hover {  background-color:
#ddd;  color: black; }
.topnav-right a.active {
background-color: #565961;
color: white; } .topnav-right
{  float: right;  padding-
right:100px;
}
.login{ margin-top:-70px;
} body {

  background-color:#ffffff;
background-repeat:  no-repeat;
background-size:cover;
background-position: 0px 0px;
  }
.login{
        margin-top:100px;
} form {border: 3px solid #f1f1f1; margin-left:400px;margin-
right:400px;} input[type=text],
input[type=email],input[type=number],input[type=password] {
width: 100%;  padding: 12px 20px;  display: inline-block;
```

```
margin-bottom:18px;  border: 1px solid #ccc;  box-sizing:
border-box;
} button {  background-
color: #28272c;  color:
white;  padding: 14px 20px;
margin-bottom:8px;
 border: none;  cursor:
pointer;  width: 100%;  font-
weight:bold; } button:hover {
opacity: 0.8; } .cancelbtn {
width: auto;  padding: 10px
18px;  background-color:
#f44336;
}
.imgcontainer {  text-
align: center;  margin:
24px 0 12px 0;
}  img.avatar  {
width: 30%;  border-
radius:  50%;  }
.container  {
padding:  16px;  }
span.psw {
 float: right;
 padding-top: 16px;
 }
```

```
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {      display: block;      float:
none;
  }
  .cancelbtn {
width: 100%;
  }
}
 </style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
padding-top:1%">DR Login Page</div>
  <div class="topnav-right" style="padding-top:0.5%;">
      <a  href="{{ url_for('index')}}">Home</a>
    <a class="active" href="{{ url_for('login')}}">Login</a>
    <a  href="{{ url_for('register')}}">Register</a>
  </div>
</div>
<div id="login" class="login">
        <form action="{{url_for('afterlogin')}}" method="post">
                <div class="imgcontainer">
                        <img style="" src="https://pngimage.net/wp-
content/uploads/2018/05/agriculteur-png.png" alt="Avatar"
class="avatar">
```

```
            </div>
<div class="container">
  <input type="email" placeholder="Enter registered email ID" name="_id" required><br>
  <input type="password" placeholder="Enter Password" name="psw" required>
      <button type="submit">Login</button><br>
   {{pred}}
            </div>
       </form>
 </div>
</body>
</html>
```

Sprint 3/Application Building/Building Html Pages/logout.html

```
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>DR logout page</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link
```

```
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Merriweather' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Josefin Sans' rel='stylesheet'>
<link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'> <style>
.header {
                top:0;
margin:0px;                left: 0px;
right: 0px;                position: fixed;
background-color: #28272c;                color:
white;                box-shadow: 0px 8px
4px grey;                overflow: hidden;
    padding-left:20px;                font-
family: 'Josefin Sans';                font-size:
2vw;                width: 100%;
                height:8%;
                text-align: center;
            }
            .topnav {
  overflow: hidden;   background-
color: #333;
}
.topnav-right a {   float:
left;      color:  #f2f2f2;
```

```
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 18px; }
.topnav-right a:hover {
background-color: #ddd;
color: black; }
.topnav-right a.active {
background-color: #565961;
color: white; } .topnav-right
{ float: right; padding-
right:100px;

}
.login{ margin-top:-70px;
} body { background-
color:#ffffff; background-
repeat: no-repeat; background-
size:cover; background-
position: 0px 0px;
  }
.main{ margin-
top:100px; text-
align:center;
} form { margin-left:400px;margin-
right:400px;}
input[type=text],
```

```
input[type=email],input[type=number],input[type=password] {
width: 100%;   padding: 12px 20px;   display: inline-block;
margin-bottom:18px;   border: 1px solid #ccc;   box-sizing:
border-box;
} button {   background-
color: #28272c;   color:
white;   padding: 14px 20px;
margin-bottom:8px;   border:
none;   cursor: pointer;
width: 20%; } button:hover {
opacity: 0.8; } .cancelbtn {
width: auto;   padding: 10px
18px;   background-color:
#f44336;
}
.imgcontainer  {      text-
align:  center;     margin:
24px 0 12px 0;
}      img.avatar      {
width: 30%;   border-
radius:      50%;      }
.container               {
padding: 16px;
}
span.psw {
  float: right;   padding-
top: 16px;
```

```
  }
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {      display: block;      float:
none;
  }
  .cancelbtn {
width: 100%;
  }
}
  </style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
padding-top:1%">DR Logout</div>
  <div class="topnav-right" style="padding-top:0.5%;">
       <a  href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('login')}}">Login</a>
    <a href="{{ url_for('register')}}">Register</a>
  </div>
</div>
<div class="main">
<h1>Successfully Logged Out!</h1>
<h3 style="color:#4CAF50">Login for more information<h3>
<a href="{{ url_for('login') }}"><button
type="submit">Login</button></a>
```

</form>

</body>

</html>

3.4.2.3 Sprint 3/Application Building/Building Html Pages/prediction.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initialscale=1.0">

<!--Bootstrap -->

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-

```
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+
76PVCmYl" crossorigin="anonymous"></script>

<script src="https://kit.fontawesome.com/8b9cdc2059.js"
crossorigin="anonymous"></script>

   <link
href="https://fonts.googleapis.com/css2?family=Akronim&family=Roboto
&display=swap" rel="stylesheet">

   <link rel="stylesheet" href="../static/style.css">

   <script defer src="../static/js/JScript.js"></script>

   <title>Prediction</title>

</head>

<body>

   <header id="head" class="header">

      <section  id="navbar">

            <h1 class="nav-heading"></i>Diabetic Retinopathy
Classification</h1>

         <div class="nav--items">

            <ul>

               <li><a  href="{{ url_for('index')}}">Home</a></li>

                              <li><a href="{{
url_for('logout')}}">Logout</a></li>

            <!-- <li><a href="#about">About</a></li>

            <li><a href="#services">Services</a></li> -->

               </ul>

         </div>

      </section>

   </header>

   <!-- dataset/Training/metal/metal326.jpg -->
```

```
   <section id="prediction">
      <div class="prediction-input">
      <div class="circle">
         <img src="../static/img/dummy/dr.jpg" alt="Demo" id="demo"
class="circle">
      </div>
            <form id="form" action="/result" method="post"
enctype="multipart/form-data">
               <input type="file" id="imageupload"  name="image"
accept="image/*" class="input-image">
               <input type="submit" class="submitbtn">
            </form>
         </div>
         <h5 class="title text-muted">
         Diabetic Retinopathy Classification is  : <b>{{prediction}}<b>
        </h5>
   <div class="line"></div>
               <div class="card" style="width: 78rem;">
               <img src="../static/img/testimonials/stages.png" alt="">
</div>
    </section>
   <section id="footer">
      <p>Copyright © 2021. All Rights Reserved</p>
       </section>
</body>
</html>
```

Sprint 3/Application Building/Building Html Pages/register.html

```
<!DOCTYPE html>

<html >

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <title> Diabetic Retinopathy Prediction</title>

  <link href='https://fonts.googleapis.com/css?family=Pacifico'
rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet'
type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300'
rel='stylesheet' type='text/css'>

<link
href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:30
0' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css')
}}">

<link href='https://fonts.googleapis.com/css?family=Merriweather'
rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Josefin Sans'
rel='stylesheet'>

<link href='https://fonts.googleapis.com/css?family=Montserrat'
rel='stylesheet'> <style>

.header {

                    top:0;

margin:0px;                    left: 0px;

right: 0px;                position: fixed;

background-color: #28272c;                color:
```

```
white;                box-shadow: 0px 8px
4px grey;             overflow: hidden;
    padding-left:20px;              font-
family: 'Josefin Sans';          font-size:
2vw;                  width: 100%;
height:8%;
                  text-align: center;
            }
            .topnav {
  overflow: hidden;  background-
color: #333;
}
.topnav-right   a   {
float:  left;   color:
#f2f2f2;   text-align:
center;  padding: 14px
16px;          text-
decoration:    none;
font-size:  18px;  }
.topnav-right  a:hover
{   background-color:
#ddd;  color: black; }
.topnav-right a.active {
background-color: #565961;
color: white; } .topnav-right
{  float: right;  padding-
right:100px;
```

```
}
.login{    margin-top:-70px;    }
body    {              background-
color:#ffffff;         background-
repeat: no-repeat;  background-
size:cover;            background-
position: 0px 0px;   }

.login{
        margin-top:100px;
} form {border: 3px solid #f1f1f1; margin-left:400px;margin-
right:400px;} input[type=text],
input[type=email],input[type=number],input[type=password] {
width: 100%;   padding: 12px 20px;   display: inline-block;
margin-bottom:18px;   border: 1px solid #ccc;   box-sizing:
border-box;
}

button {   background-color:
#28272c;   color: white;
padding: 14px 20px;
margin-bottom:8px;   border:
none;   cursor: pointer;
width: 100%;
} button:hover
{   opacity:
0.8; }
```

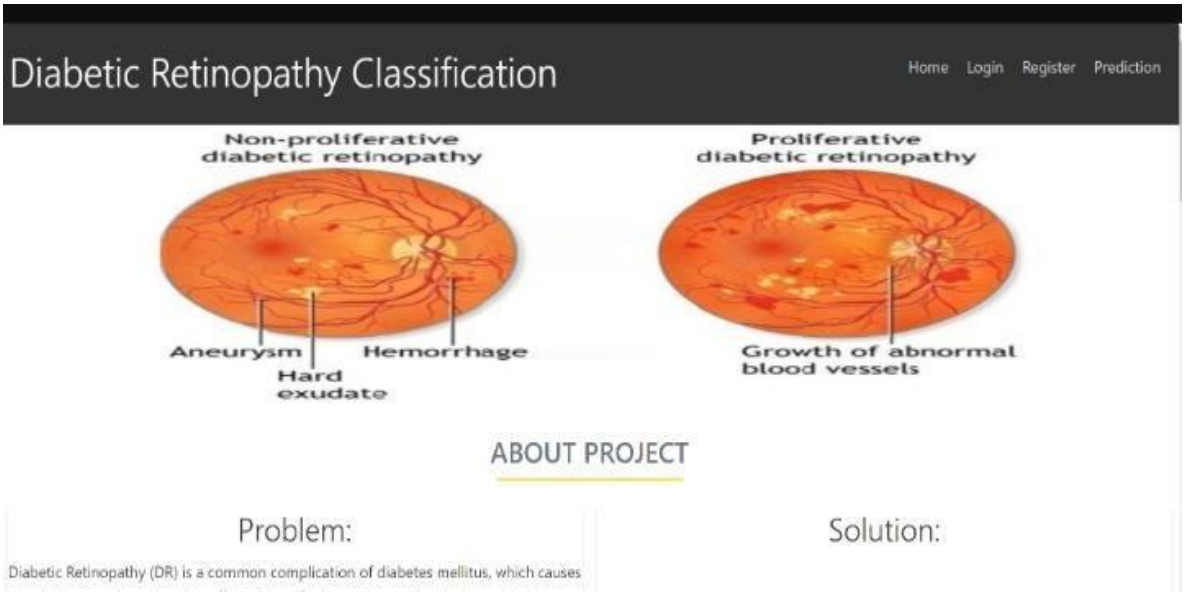
```
.cancelbtn {      width:  auto;
padding:       10px       18px;
background-color: #f44336;
}
.imgcontainer  {      text-
align:  center;     margin:
24px 0 12px 0;
}      img.avatar      {
width: 30%;   border-
radius: 50%;
}
.container {
padding: 16px;
} span.psw
{
  float: right;   padding-
top: 16px;
  }
/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
span.psw {      display: block;      float:
none;
  }
  .cancelbtn {
width: 100%;
  }
```

```
}
  </style>
</head>
<body style="font-family:Montserrat;">
<div class="header">
 <div style="width:50%;float:left;font-size:2vw;text-align:left;color:white;
padding-top:1%">DR Registration</div>
  <div class="topnav-right" >
       <a  href="{{ url_for('home')}}">Home</a>
    <a href="{{ url_for('login')}}">Login</a>
 <a class="active"  href="{{ url_for('register')}}">Register</a>
  </div>
</div>
<div id="login" class="login">
        <form action="{{url_for('afterreg')}}" method="post">
              <div class="imgcontainer">
                    <img style=""
src="https://www.iconbunny.com/icons/media/catalog/product/2/1/2136.1
2-farmer-icon-iconbunny.jpg" alt="Avatar" class="avatar">
              </div>
<div class="container">
                    <input type="text" placeholder="Enter Name"
name="name" required><br>
   <input type="email" placeholder="Enter Email ID" name="_id"
required><br>
   <input type="password" placeholder="Enter Password" name="psw"
required>
       <button type="submit">Register</button><br>
```

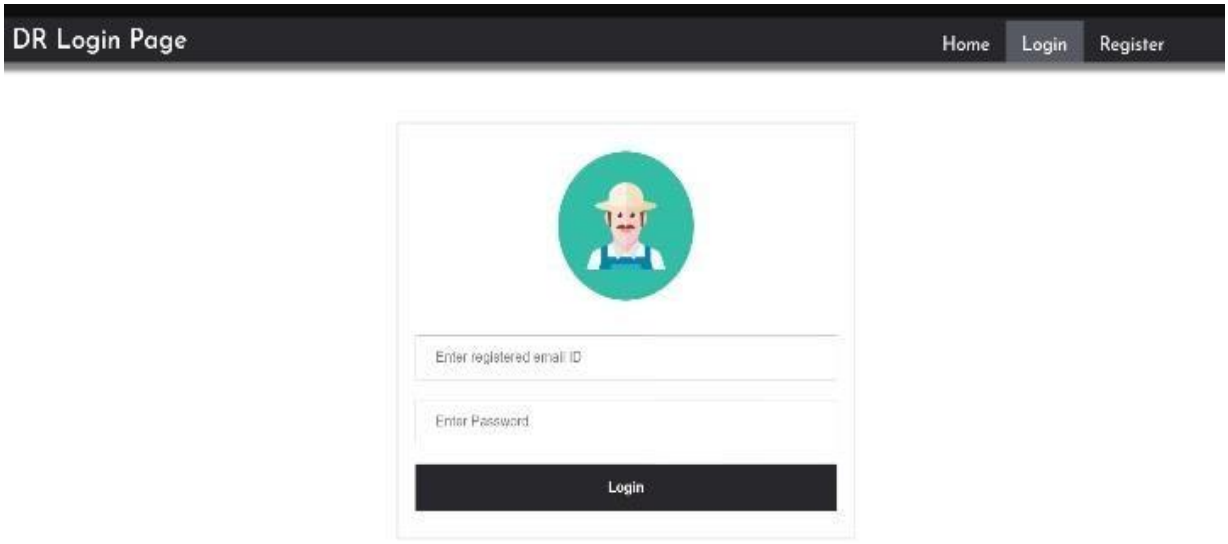
```
    {{pred}}
                </div>
                <div class="container" style="background-color:#f1f1f1">
    <div class="psw">Already have an account?&amp;nbsp; &amp;nbsp;<a href="{{
url_for('login') }}">Login</a></div>
  </div>
        </form>
</div>
 </body>
</html>
```

Sprint 3/Application Building/Run The Application/

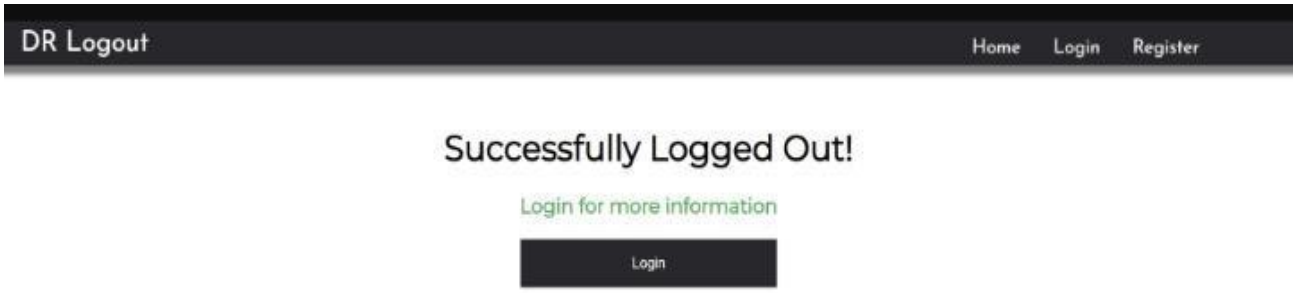
3.4.2.4 Sprint 3/Application Building/Run The Application/Home Page.png



3.4.2.5 Sprint 3/Application Building/Run The Application/Login page.png

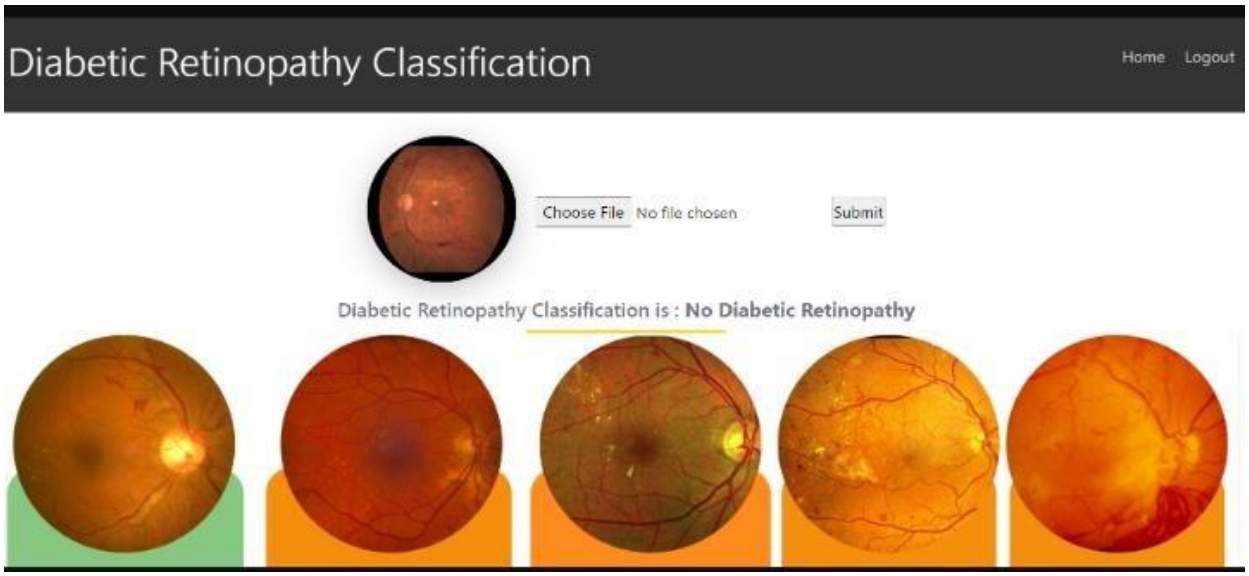


3.4.2.6 Sprint 3/Application Building/Run The Application/Logout Page.png

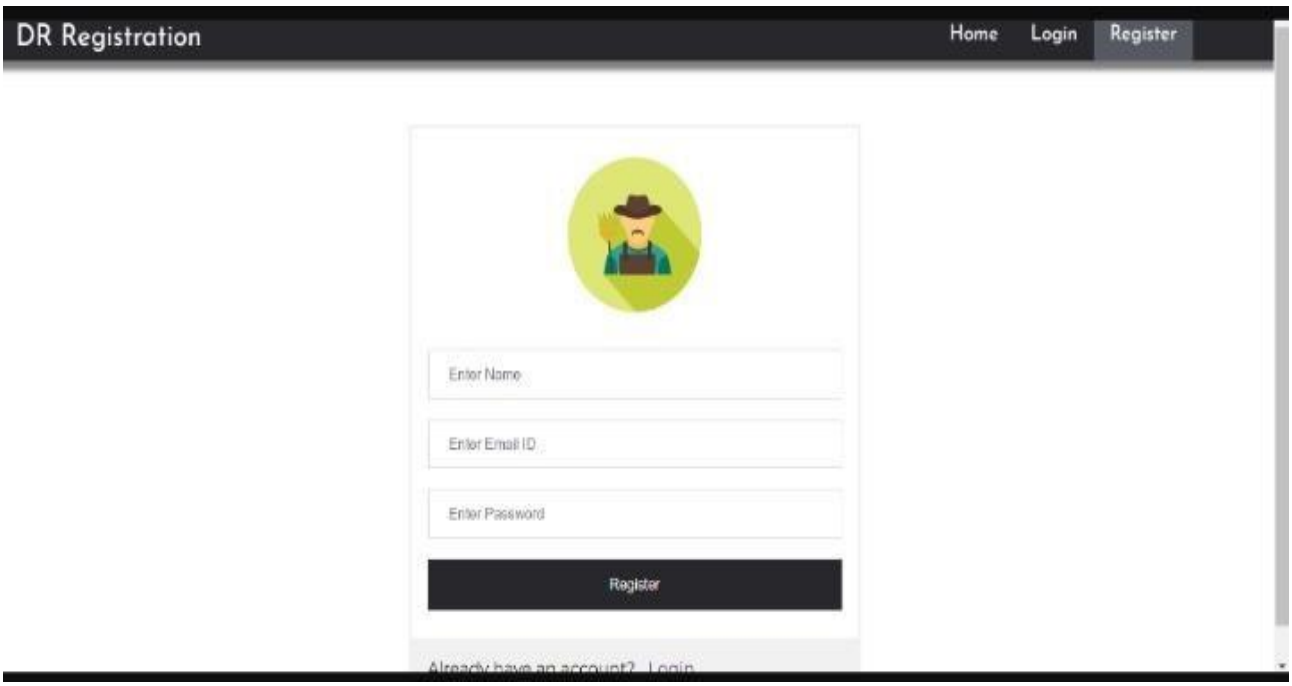


3.4.2.7 Sprint 3/Application Building/Run The

Application/Prediction Page.png

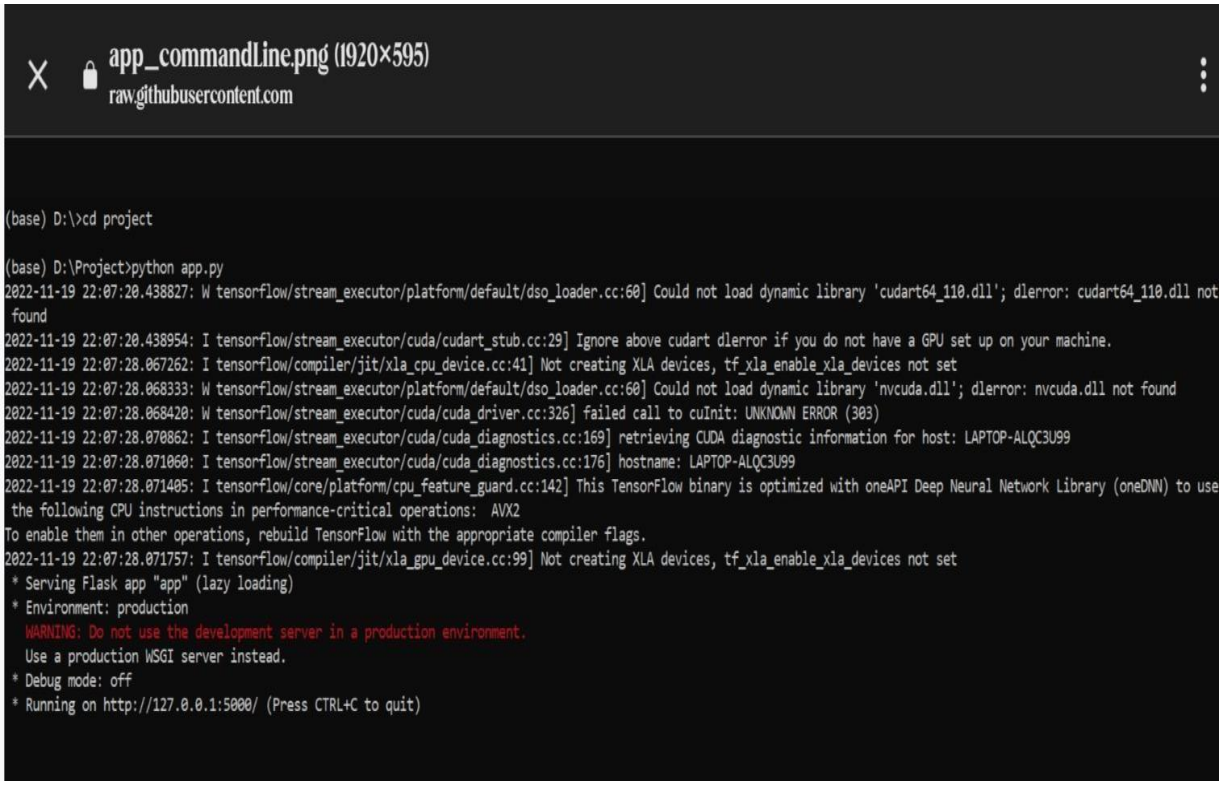


3.4.2.8 Sprint 3/Application Building/Run The Application/Registration page.png

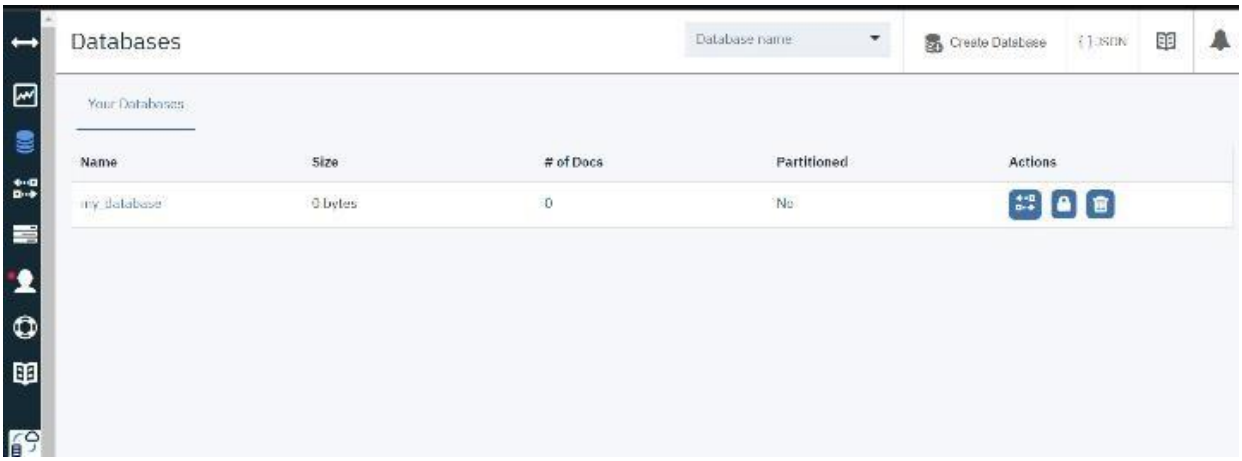


3.4.2.9 Sprint 3/Application Building/Run The

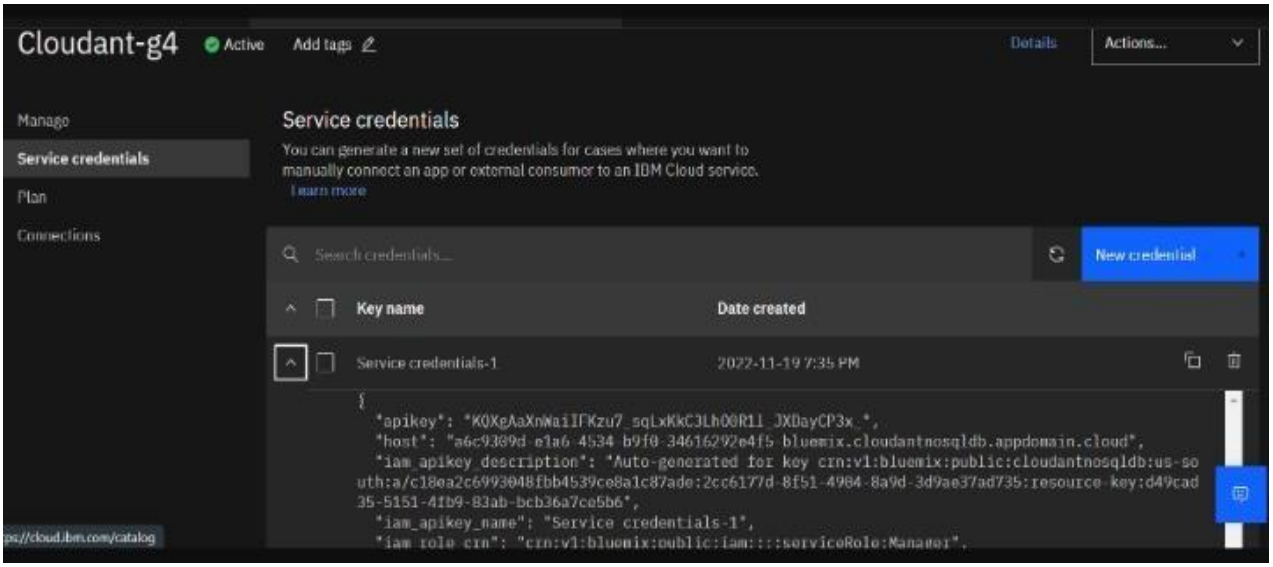
Application/app_commandLine.png



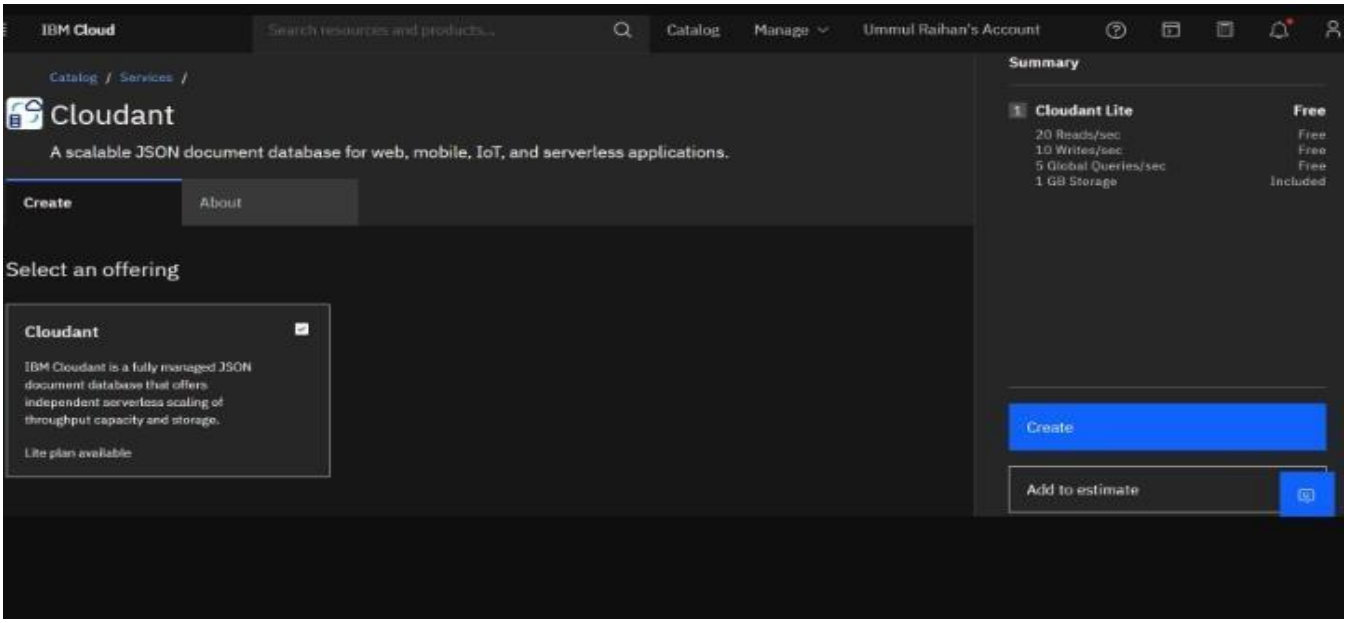
Create cloudant DB Sprint 3/Create Cloudant DB/Create Database.png



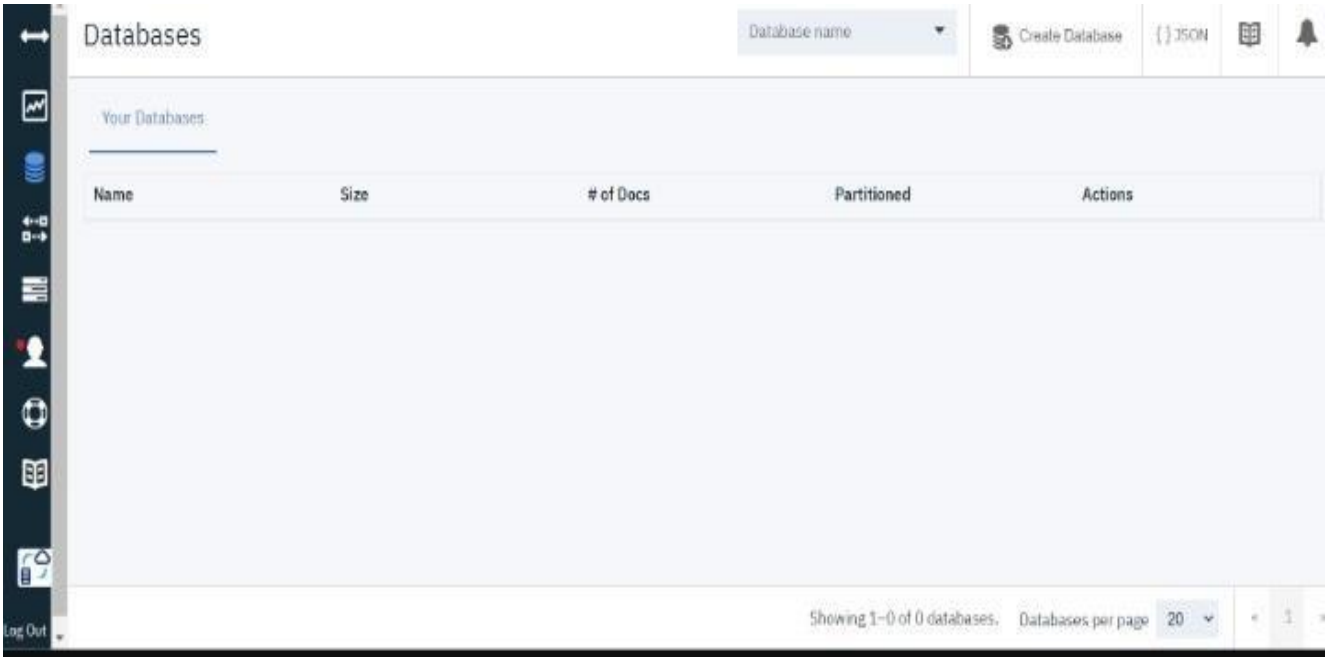
Sprint 3/Create Cloudant DB/Create Service Credential.png



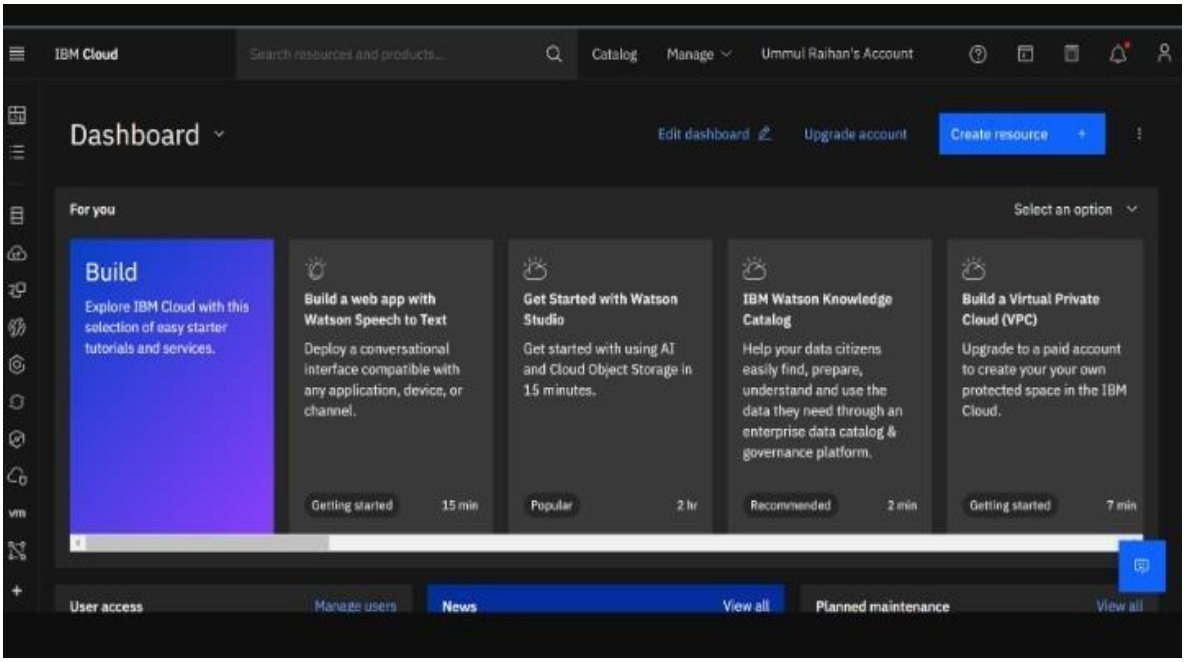
Sprint 3/Create Cloudant DB/Create Service Instance.png



Sprint 3/Create Cloudant DB/Launch Database.png

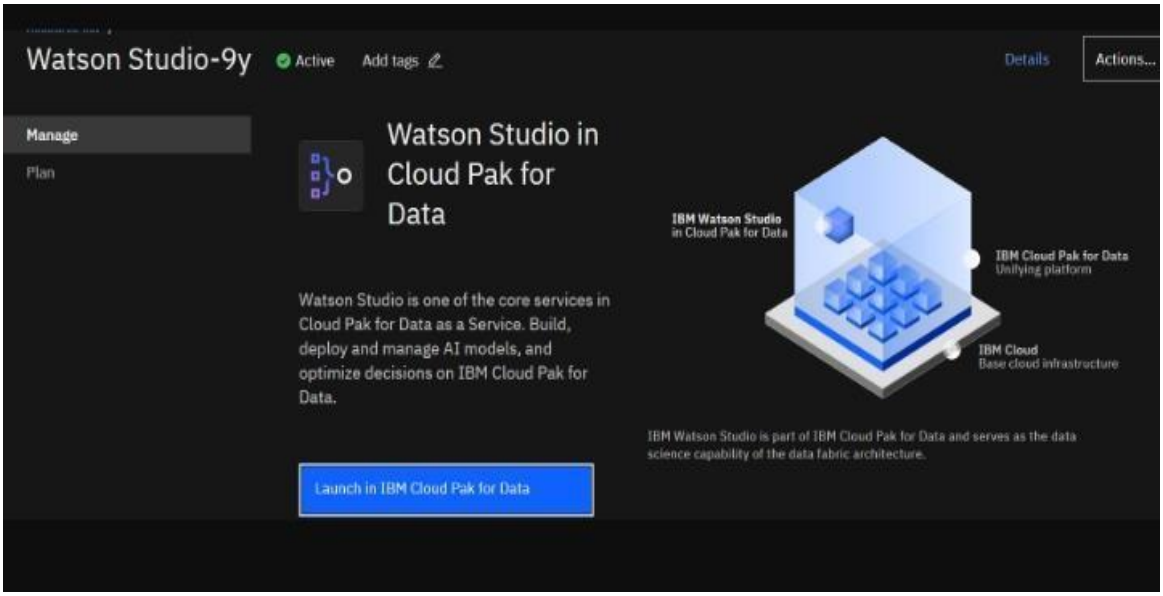


Sprint 3/Create Cloudant DB/Register & Login to IBM Cloud.png



SPRINT-4

Sprint 4/Train the Model on IBM/ Sprint 4/Train the Model on IBM/Watson Studio.png



3.5 Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	`[16] model.summary()`

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 299, 299, 3)]     0         
block1_conv1 (Conv2D)        (None, 149, 149, 32)      864       
block1_conv1_bn (BatchNormaliz (None, 149, 149, 32)    128       
ation)                                                           
block1_conv1_act (Activation) (None, 149, 149, 32)     0         
block1_conv2 (Conv2D)        (None, 147, 147, 64)      18432     
block1_conv2_bn (BatchNormaliz (None, 147, 147, 64)    256       
ation)                                                           
block1_conv2_act (Activation) (None, 147, 147, 64)     0         
block2_sepconv1 (SeparableConv (None, 147, 147, 128)   8768      
2D)                                                              
block2_sepconv1_bn (BatchNorma (None, 147, 147, 128)   512       
lization)                                                        
block2_sepconv2_act (Activatio (None, 147, 147, 128)   0         
n)                                                               
```

```
block2_sepconv2 (SeparableConv  (None, 147, 147, 12  17536       ['block2_sepconv2_act[0][0]']
2D)                             8)

block2_sepconv2_bn (BatchNorma  (None, 147, 147, 12  512         ['block2_sepconv2[0][0]']
lization)                       8)

conv2d (Conv2D)                 (None, 74, 74, 128)  8192        ['block1_conv2_act[0][0]']

block2_pool (MaxPooling2D)      (None, 74, 74, 128)  0           ['block2_sepconv2_bn[0][0]']

batch_normalization (BatchNorm  (None, 74, 74, 128)  512         ['conv2d[0][0]']
alization)

add (Add)                       (None, 74, 74, 128)  0           ['block2_pool[0][0]',
                                                                 'batch_normalization[0][0]']

block3_sepconv1_act (Activatio  (None, 74, 74, 128)  0           ['add[0][0]']
n)

block3_sepconv1 (SeparableConv  (None, 74, 74, 256)  33920       ['block3_sepconv1_act[0][0]']
2D)

block3_sepconv1_bn (BatchNorma  (None, 74, 74, 256)  1024        ['block3_sepconv1[0][0]']
lization)

block3_sepconv2_act (Activatio  (None, 74, 74, 256)  0           ['block3_sepconv1_bn[0][0]']
n)

block3_sepconv2 (SeparableConv  (None, 74, 74, 256)  67840       ['block3_sepconv2_act[0][0]']
2D)

block3_sepconv2_bn (BatchNorma  (None, 74, 74, 256)  1024        ['block3_sepconv2[0][0]']
lization)

conv2d_1 (Conv2D)               (None, 37, 37, 256)  32768       ['add[0][0]']

block3_pool (MaxPooling2D)      (None, 37, 37, 256)  0           ['block3_sepconv2_bn[0][0]']

batch_normalization_1 (BatchNo  (None, 37, 37, 256)  1024        ['conv2d_1[0][0]']
rmalization)

add_1 (Add)                     (None, 37, 37, 256)  0           ['block3_pool[0][0]',
                                                                 'batch_normalization_1[0][0]']

block4_sepconv1_act (Activatio  (None, 37, 37, 256)  0           ['add_1[0][0]']
n)

block4_sepconv1 (SeparableConv  (None, 37, 37, 728)  188672      ['block4_sepconv1_act[0][0]']
2D)

block4_sepconv1_bn (BatchNorma  (None, 37, 37, 728)  2912        ['block4_sepconv1[0][0]']
lization)

block4_sepconv2_act (Activatio  (None, 37, 37, 728)  0           ['block4_sepconv1_bn[0][0]']
n)

block4_sepconv2 (SeparableConv  (None, 37, 37, 728)  536536      ['block4_sepconv2_act[0][0]']
2D)

block4_sepconv2_bn (BatchNorma  (None, 37, 37, 728)  2912        ['block4_sepconv2[0][0]']
lization)

conv2d_2 (Conv2D)               (None, 19, 19, 728)  186368      ['add_1[0][0]']

block4_pool (MaxPooling2D)      (None, 19, 19, 728)  0           ['block4_sepconv2_bn[0][0]']

batch_normalization_2 (BatchNo  (None, 19, 19, 728)  2912        ['conv2d_2[0][0]']
rmalization)

add_2 (Add)                     (None, 19, 19, 728)  0           ['block4_pool[0][0]',
                                                                 'batch_normalization_2[0][0]']

block5_sepconv1_act (Activatio  (None, 19, 19, 728)  0           ['add_2[0][0]']
n)

block5_sepconv1 (SeparableConv  (None, 19, 19, 728)  536536      ['block5_sepconv1_act[0][0]']
2D)

block5_sepconv1_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block5_sepconv1[0][0]']
lization)

block5_sepconv2_act (Activatio  (None, 19, 19, 728)  0           ['block5_sepconv1_bn[0][0]']
n)

block5_sepconv2 (SeparableConv  (None, 19, 19, 728)  536536      ['block5_sepconv2_act[0][0]']
2D)
```

```
block6_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block6_sepconv2[0][0]']
lization)

block6_sepconv3_act (Activatio  (None, 19, 19, 728)  0          ['block6_sepconv2_bn[0][0]']
n)

block6_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536     ['block6_sepconv3_act[0][0]']
2D)

block6_sepconv3_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block6_sepconv3[0][0]']
lization)

add_4 (Add)                     (None, 19, 19, 728)  0           ['block6_sepconv3_bn[0][0]',
                                                                 'add_3[0][0]']

block7_sepconv1_act (Activatio  (None, 19, 19, 728)  0          ['add_4[0][0]']
n)

block7_sepconv1 (SeparableConv  (None, 19, 19, 728)  536536     ['block7_sepconv1_act[0][0]']
2D)

block7_sepconv1_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block7_sepconv1[0][0]']
lization)

block7_sepconv2_act (Activatio  (None, 19, 19, 728)  0          ['block7_sepconv1_bn[0][0]']
n)

block7_sepconv2 (SeparableConv  (None, 19, 19, 728)  536536     ['block7_sepconv2_act[0][0]']
2D)

block7_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block7_sepconv2[0][0]']
lization)

block7_sepconv3_act (Activatio  (None, 19, 19, 728)  0          ['block7_sepconv2_bn[0][0]']
n)

block7_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536     ['block7_sepconv3_act[0][0]']
2D)

block7_sepconv3_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block7_sepconv3[0][0]']
lization)

add_5 (Add)                     (None, 19, 19, 728)  0           ['block7_sepconv3_bn[0][0]',
                                                                 'add_4[0][0]']
```

```
block5_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block5_sepconv2[0][0]']
lization)

block5_sepconv3_act (Activatio  (None, 19, 19, 728)  0          ['block5_sepconv2_bn[0][0]']
n)

block5_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536     ['block5_sepconv3_act[0][0]']
2D)

block5_sepconv3_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block5_sepconv3[0][0]']
lization)

add_3 (Add)                     (None, 19, 19, 728)  0           ['block5_sepconv3_bn[0][0]',
                                                                 'add_2[0][0]']

block6_sepconv1_act (Activatio  (None, 19, 19, 728)  0          ['add_3[0][0]']
n)

block6_sepconv1 (SeparableConv  (None, 19, 19, 728)  536536     ['block6_sepconv1_act[0][0]']
2D)

block6_sepconv1_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block6_sepconv1[0][0]']
lization)

block6_sepconv2_act (Activatio  (None, 19, 19, 728)  0          ['block6_sepconv1_bn[0][0]']
n)

block6_sepconv2 (SeparableConv  (None, 19, 19, 728)  536536     ['block6_sepconv2_act[0][0]']
2D)

block6_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912        ['block6_sepconv2[0][0]']
lization)

block6_sepconv3_act (Activatio  (None, 19, 19, 728)  0          ['block6_sepconv2_bn[0][0]']
n)

block6_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536     ['block6_sepconv3_act[0][0]']
2D)
```

```
block8_sepconv1_act (Activatio  (None, 19, 19, 728)  0          ['add_5[0][0]']
n)

block8_sepconv1 (SeparableConv  (None, 19, 19, 728)  536536     ['block8_sepconv1_act[0][0]']
2D)

block8_sepconv1_bn (BatchNorma  (None, 19, 19, 728)  2912       ['block8_sepconv1[0][0]']
lization)

block8_sepconv2_act (Activatio  (None, 19, 19, 728)  0          ['block8_sepconv1_bn[0][0]']
n)

block8_sepconv2 (SeparableConv  (None, 19, 19, 728)  536536     ['block8_sepconv2_act[0][0]']
2D)

block8_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912       ['block8_sepconv2[0][0]']
lization)

block8_sepconv3_act (Activatio  (None, 19, 19, 728)  0          ['block8_sepconv2_bn[0][0]']
n)
```

.
.
.

```
 add_11 (Add)                   (None, 10, 10, 1024  0          ['block13_pool[0][0]',
                                )                                 'batch_normalization_3[0][0]']

 block14_sepconv1 (SeparableCon  (None, 10, 10, 1536  1582080    ['add_11[0][0]']
 v2D)                           )

 block14_sepconv1_bn (BatchNorm  (None, 10, 10, 1536  6144       ['block14_sepconv1[0][0]']
 alization)                     )

 block14_sepconv1_act (Activati  (None, 10, 10, 1536  0          ['block14_sepconv1_bn[0][0]']
 on)                            )

 block14_sepconv2 (SeparableCon  (None, 10, 10, 2048  3159552    ['block14_sepconv1_act[0][0]']
 v2D)                           )

 block14_sepconv2_bn (BatchNorm  (None, 10, 10, 2048  8192       ['block14_sepconv2[0][0]']
 alization)                     )

block14_sepconv2_act (Activati  (None, 10, 10, 2048  0          ['block14_sepconv2_bn[0][0]']
on)                            )

flatten (Flatten)               (None, 204800)       0          ['block14_sepconv2_act[0][0]']

dense (Dense)                   (None, 5)            1024005    ['flatten[0][0]']

==================================================================================================
Total params: 21,885,485
Trainable params: 1,024,005
Non-trainable params: 20,861,480
__________________________________________________________________________________________________
```

2.	Accuracy	Training Accuracy - Validation Accuracy -	```# fit the model r = model.fit_generator(training_set, validation_data=test_set, epochs=30, steps_per_epoch=len (training_set)//32, validation_steps=len(test_set)//32)``` ```Epoch 1/30 3/3 [==============================] - 58s 17s/step - loss: 12.1428 - accuracy: 0.3229 Epoch 2/30 3/3 [==============================] - 50s 14s/step - loss: 10.8191 - accuracy: 0.5521 Epoch 3/30 3/3 [==============================] - 51s 16s/step - loss: 9.6766 - accuracy: 0.4688 Epoch 4/30 3/3 [==============================] - 51s 16s/step - loss: 7.3417 - accuracy: 0.5833 Epoch 5/30 3/3 [==============================] - 49s 14s/step - loss: 5.9892 - accuracy: 0.5208 Epoch 6/30 3/3 [==============================] - 47s 14s/step - loss: 4.0807 - accuracy: 0.6771 Epoch 7/30 3/3 [==============================] - 49s 15s/step - loss: 3.9948 - accuracy: 0.6562 Epoch 8/30 3/3 [==============================] - 49s 15s/step - loss: 4.0479 - accuracy: 0.6250 Epoch 9/30 3/3 [==============================] - 50s 15s/step - loss: 4.3574 - accuracy: 0.6458 Epoch 10/30 3/3 [==============================] - 50s 15s/step - loss: 3.7197 - accuracy: 0.6146 Epoch 11/30 3/3 [==============================] - 47s 14s/step - loss: 5.1180 - accuracy: 0.5625```
			``Epoch 12/30 3/3 [==============================] - 48s 14s/step - loss: 2.5951 - accuracy: 0.7188 Epoch 13/30 3/3 [==============================] - 51s 15s/step - loss: 3.6282 - accuracy: 0.7083 Epoch 14/30 3/3 [==============================] - 47s 14s/step - loss: 3.2756 - accuracy: 0.7083 Epoch 15/30 3/3 [==============================] - 40s 15s/step - loss: 4.7868 - accuracy: 0.6795 Epoch 16/30 3/3 [==============================] - 49s 14s/step - loss: 2.7478 - accuracy: 0.7604 Epoch 17/30 3/3 [==============================] - 47s 14s/step - loss: 4.2101 - accuracy: 0.5417 Epoch 18/30 3/3 [==============================] - 48s 14s/step - loss: 4.3796 - accuracy: 0.6875 Epoch 19/30 3/3 [==============================] - 54s 17s/step - loss: 5.3032 - accuracy: 0.5312 Epoch 20/30 3/3 [==============================] - 50s 15s/step - loss: 3.7652 - accuracy: 0.7083 Epoch 21/30 3/3 [==============================] - 48s 14s/step - loss: 2.8421 - accuracy: 0.7812 Epoch 22/30 3/3 [==============================] - 48s 15s/step - loss: 2.7402 - accuracy: 0.6979 Epoch 23/30 3/3 [==============================] - 49s 15s/step - loss: 2.7817 - accuracy: 0.6771 Epoch 24/30 3/3 [==============================] - 49s 15s/step - loss: 3.3278 - accuracy: 0.7083 Epoch 25/30 3/3 [==============================] - 49s 14s/step - loss: 3.9974 - accuracy: 0.6354 Epoch 26/30 3/3 [==============================] - 48s 14s/step - loss: 2.6000 - accuracy: 0.6979 Epoch 27/30 3/3 [==============================] - 48s 15s/step - loss: 3.0479 - accuracy: 0.6979 Epoch 28/30 3/3 [==============================] - 47s 14s/step - loss: 1.9773 - accuracy: 0.7708 Epoch 29/30 3/3 [==============================] - 49s 14s/step - loss: 2.6960 - accuracy: 0.7292 Epoch 30/30 3/3 [==============================] - 47s 14s/step - loss: 2.5824 - accuracy: 0.7708```
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	NA

3.6 ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	4	1	0	0	5
Duplicate	4	1	3	2	10
External	1	3	2	0	6
Fixed	2	4	4	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	11	9	9	5	34

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

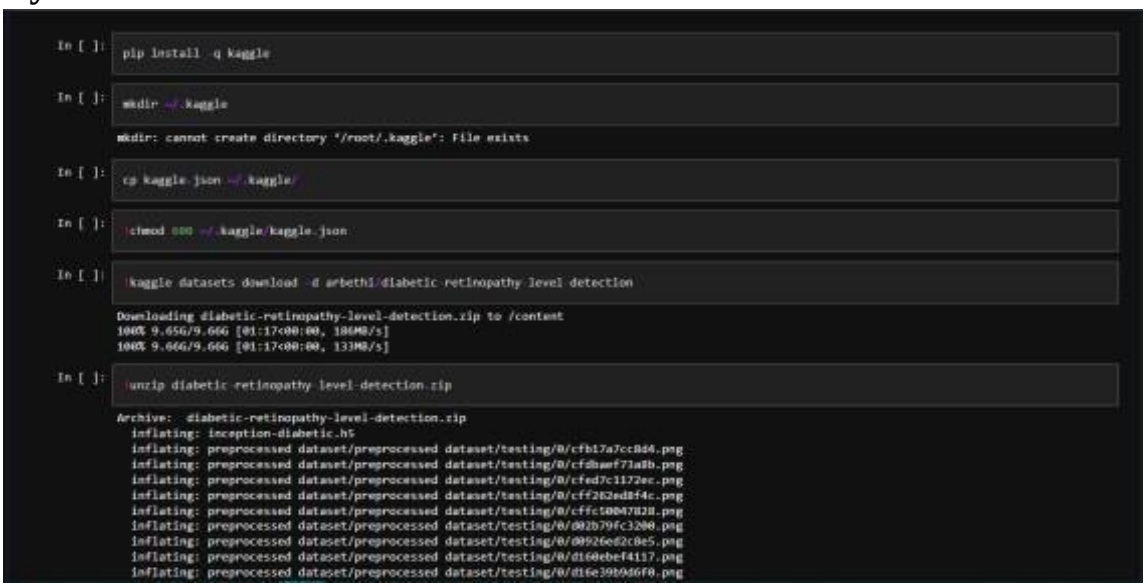
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	0	6
Client Application	12	1	0	11
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	6	0	0	6
Final Report Output	4	0	0	4
Version Control	2	0	0	2

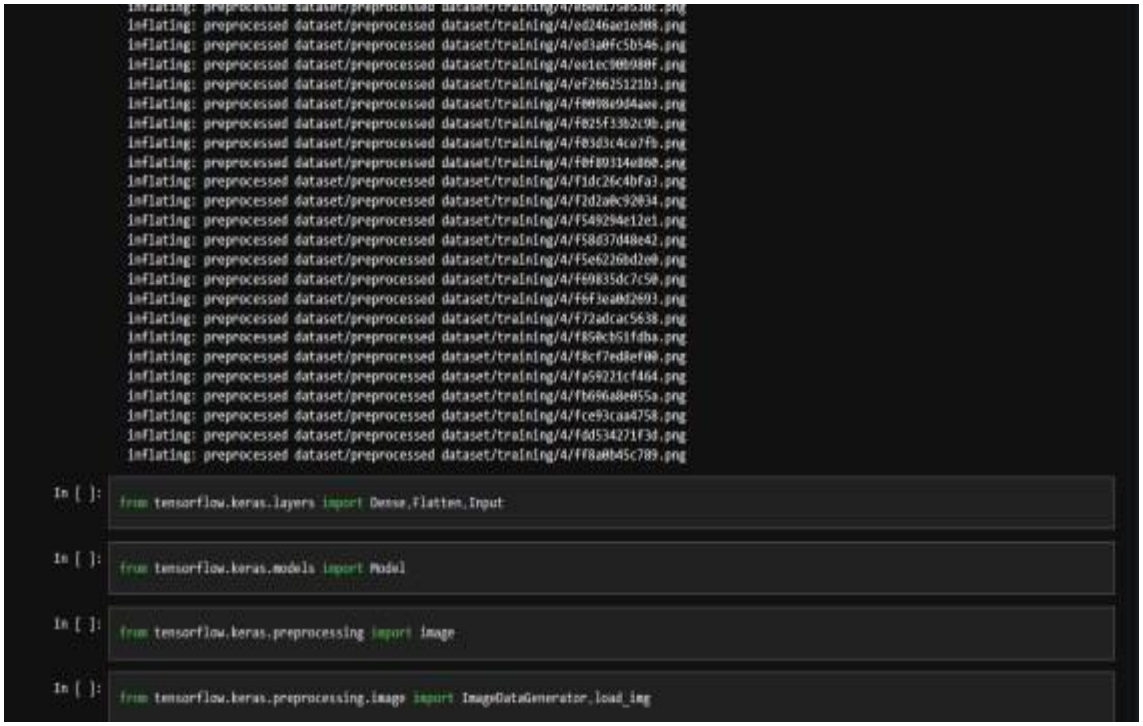
4. Final Deliverables

IBM LINK: https://github.com/IBM-EPBL/IBM-Project-192511659694823

DEMOLINK:https://drive.google.com/file/d/1Aimbmb9MbtOGrFeuKYHef9nsCcILp4xj/view?usp=sharing

5. Python Notebook screenshots:-





In [ ]:
```
from glob import glob
```

In [ ]:
```
import numpy as np
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
ImageSize=[299,299]
```

In [ ]:
```
trainPath=r"/content/preprocessed dataset/preprocessed dataset/training"
```

In [ ]:
```
testPath=r"/content/preprocessed dataset/preprocessed dataset/testing"
```

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

In [ ]:
```
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [ ]:
```
training_set=train_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/training',target_size=(299,299),batch_size=
```

```
Found 3662 images belonging to 5 classes.
```

In [ ]:
```
test_set=test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',target_size=(299,299),batch_size=32,cla
```

```
Found 3662 images belonging to 5 classes.
```

In [ ]:
```
test_set=test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',target_size=(299,299),batch_size=3
```

```
Found 734 images belonging to 5 classes.
```

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

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_no
5
83683744/83683744 [==============================] - 0s 0us/step
```

In [ ]:
```
for layer in xception.layers:
  layer.trainable=False
```

In [ ]:
```
x=Flatten()(xception.output)
```

In [ ]:
```
prediction=Dense(5,activation='softmax')(x)
```

In [ ]:
```
model=Model(inputs=xception.input,outputs=prediction)
```

In [ ]:
```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 299, 299, 3       0         
                             )]                                  
```

6. ADVANTAGES

A deep learning system could increase the cost-effectiveness of screening and diagnosis, while attaining higher than recommended performance, and that the system could be applied in clinical examinations requiring finer grading.

deep learning is well-suited for image analysis tasks. This is because deep learning algorithms can automatically learn features from images, which is essential for accurate image analysis. It is efficient at handling large amounts of data. This is important for medical image analysis, as medical images are often very large. It is scalable. This means that it can be used to train models on very large datasets, which is important for medical image analysis tasks where data is often limited. It is able to learn from data with little supervision. This is important for medical image analysis, as often there is limited labeled data available.It is robust. This means that it is less likely to overfit to the data, which is important for medical image analysis where data is often limited.

7. DISADVANTAGES:

Deep learning requires a large amount of data to train the models. This can be a challenge for researchers who do not have access to a large dataset. Another challenge is that deep learning models can be very complex, which can make them difficult to interpret. Finally, deep learning models can be computationally intensive, which can make them difficult to deploy in resource-limited settings.

8. CONCLUSION:-

Diabetic retinopathy (DR) is a leading cause of blindness in the United States. Early detection and treatment of DR is critical to preventing vision loss. However, DR is often asymptomatic in its early stages, making it difficult to detect.

Deep learning (DL) is a type of artificial intelligence that can be used to automatically detect patterns in data. DL has been shown to be effective for detecting DR in images of the retina.In this study, a DL algorithm was used to automatically detect DR in fundus images. The algorithm was able to accurately detect DR in early stages, before it is symptomatic. This could

potentially lead to earlier diagnosis and treatment of DR, which could help to prevent vision loss.

It is a serious complication of diabetes that causes progressive retinal deterioration and, in severe cases, blindness. To avoid degeneration and retinal damage, it is vital to recognize and treat it as soon as possible. In recent years, there has been a surge in interest in employing deep learning to detect diabetic retinopathy, and as various DL systems develop and are integrated into clinical practice, clinicians will be able to treat patients more successfully and efficiently. The current state of research on the application of deep learning in the diagnosis of diabetic retinopathy is examined in this article. Ophthalmologists still need to improve their performance, interpretability, and trustworthiness, despite the fact that deep learning has paved the way for more accurate diagnosis and treatment

9.FUTURE SCOPE:-

There is a great potential for deep learning in fundus image analysis for early detection of diabetic retinopathy. However, there are a few challenges that need to be addressed. First, the current data sets are small and lack diversity. Second, the images are often low quality and need to be pre-processed before they can be used for deep learning.

Third, the ground truth labels for the images are often not available. Finally, the current deep learning models are not able to generalize well to real-world data

Github Link: https://github.com/IBM-EPBL/IBM-Project-19251-1659694823

Project Demonstration Link: https://youtu.be/eKphilipomE