PROJECT DOCUMENTATION

Deep Learning Fundus Image Analysis For Early Detection of Diabetic Retinopathy

Team ID: PNT2022TMID20760

Submitted By:-

Yashwanth Ramesh – 193002126 - Team Lead

Nitheesh – 193002068

Ramnarayanan G - 193002079

Vidyarth - 193002115

Table of Contents:-

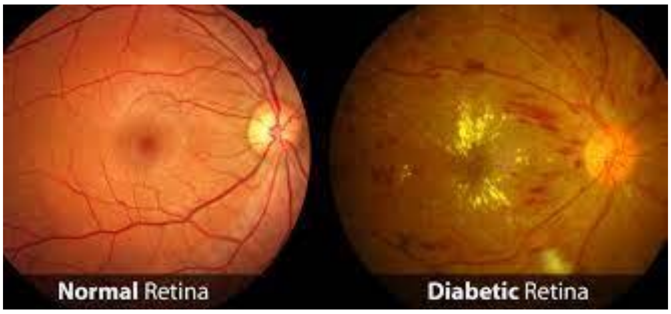
. APPENDIX

1.Source Code

2.GitHub & Project Demo Link

1. Introduction:

Diabetic retinopathy is the primary cause of vision loss worldwide. Several anomalies in the eye fundus, including microaneurysms and/or dot haemorrhages, vascular hyper permeability indications, exudates, and capillary closures, have an impact on the retinal microvasculature in the early stages of this disease. The likelihood that the laser photocoagulation required progression to the level is principally increased by the dynamics of micro-aneurysms. Lesions caused by diabetic retinopathy are typically believed to be reversible, and the illness can only grow more slowly early on. It is anticipated that identifying patients who have these initial lesions—primarily Microaneurysms and small blood cells—through recurrent examination would open up new opportunities for treating retinopathy. Common signs of diabetic retinopathy include floating and flashing, blurred vision, and sudden loss of vision.



1.1 Project Overview:

Diabetes mellitus frequently results in diabetic retinopathy (DR), which results in lesions on the retina that impair vision. Blindness may result if it is not caught in time. Unfortunately, there is no cure for DR; treatment merely preserves vision. Early diagnosis and treatment of DR can greatly lower the risk of visual loss. In contrast to computer-aided diagnosis systems, the manual diagnosis process of DR retina fundus photographs by ophthalmologists is costly, time-consuming, and prone to error.

One of the most popular methods for improving performance, particularly in the categorization and interpretation of medical images, is transfer learning. We employed transfer learning methods that are more commonly used in medical image analysis and are quite successful, such as Inception V3, Resnet50, and Xception V3. The main objective is to detect the Diabetic Retinopathy in early stages by processing the Retinal images. Transfer learning has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. We used Transfer Learning techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective. Deep Learning, Machine learning, Neural Networks and knowledge in Python will play a significant role in the development of our Project.

1.2 Purpose:

The proposed study aims to automate the crucial in ophthalmology identification and categorization of diabetic retinopathy using retinal fundus images. The majority of currently used techniques rely on manually created features, which are then supplied into the classifier for detection and classification purposes. Convolutional neural networks (CNNs) have recently been utilised to solve this classification problem, however their architecture was manually created. This study proposes a genetic algorithm-based method for automatically determining CNN parameters, which is then used to the categorization of diabetic retinopathy. The convolution and pooling layers in the proposed CNN model are used to extract features. Finally, classification is performed using support vector machines (SVM). The evolutionary algorithm is used to determine hyper-parameters such as the number of convolution and pooling layers, the number of kernels, and the size of the convolution layer kernels. The Messidor dataset, which is available to the public, is used to test the proposed methodology. The accuracy and AUC of the suggested approach were both 0.9867. The results of the experiments demonstrate that the suggested auto-tuned CNN performs noticeably better than the current approaches. Using CNN relieves the difficulty of creating image features, while on the other hand, genetic algorithm-based methodology automates CNN hyper-parameter construction.

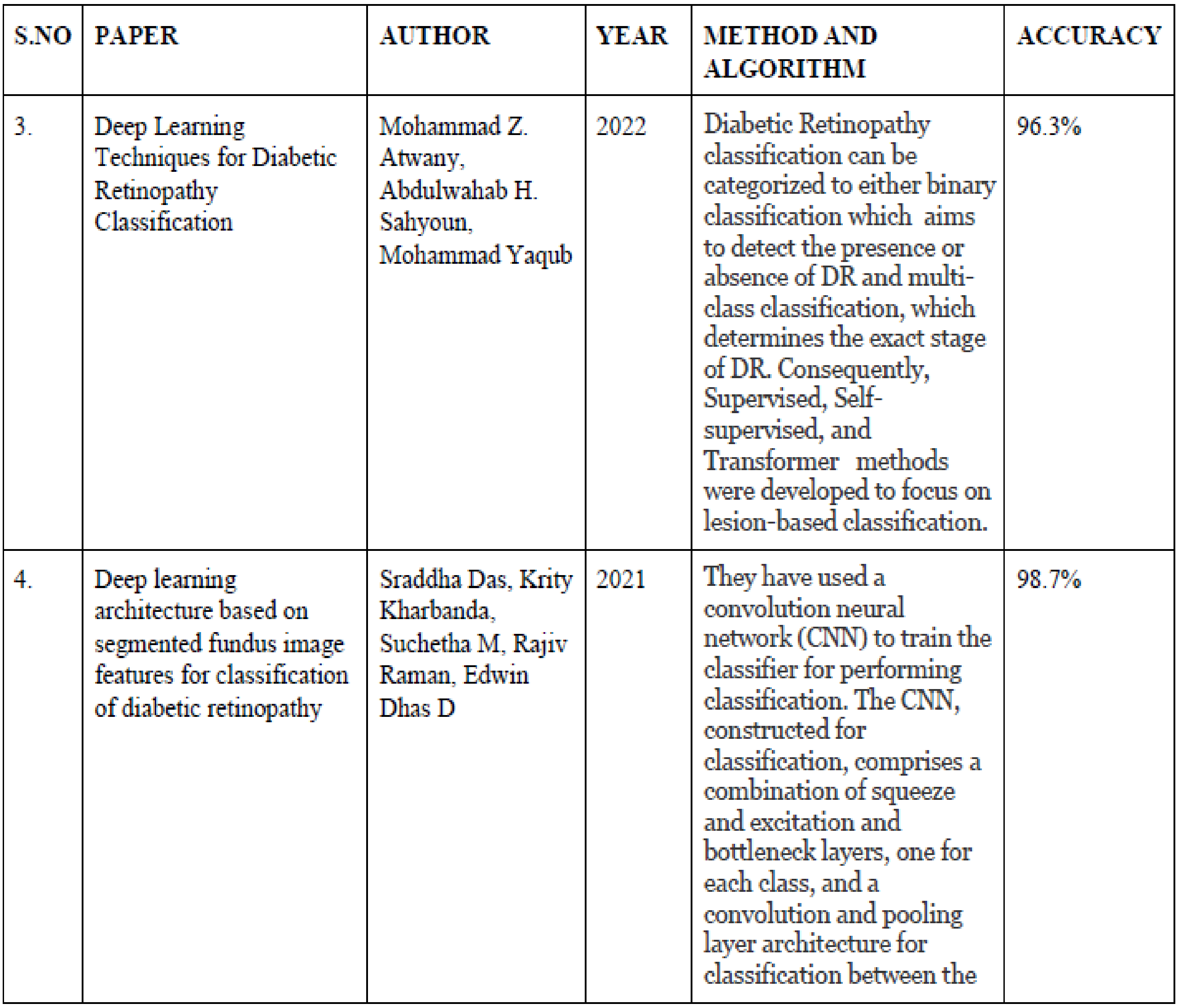
2. Literature Survey:

2.1 Existing Problem:

As a result of diabetes mellitus, which causes lesions on the retina of the eyes, diabetic retinopathy (DR) is a degenerative condition that affects the eyes. For diabetic patients, especially the working-age population in poor countries, diabetic retinopathy is thought to be the main cause of blindness. Since the condition is irreversible, treatment focuses on maintaining the patient's level of eyesight. In order to effectively maintain the patient's vision, early identification of diabetic retinopathy is essential. The biggest problem with DR detection is that manual diagnosis requires a lot of time, money, and effort and requires an ophthalmologist to examine retinal fundus images of the eyes. The latter is also more challenging, especially in the early stages of the illness when the disease's symptoms are less obvious in the photos. Retinal fundus images can be evaluated using machine learning-based medical image analysis, and the use of deep learning algorithms has facilitated the early identification of diabetic retinopathy (DR). This work proposes retinal fundus picture classification and detection using state-of-the-art deep learning techniques in supervised, self-supervised, and Vision Transformer configurations. For instance, classifications of diabetic retinopathy that are referable, nonreferable, and proliferative are evaluated and summarised.

2.2 References:

S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM	ACCURACY
1.	Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening	Tao Li, Yingqi Gao, Kai Wang, Song Guo, Hanruo Liu, Hong Kang	2019	They collected 13,673 fundus images from 9598 patients. These images were divided into six classes by seven graders according to image quality and DR level. Moreover, 757 images with DR were selected to annotate four types of DR-related lesions. Finally, we evaluated state-of-the-art deep learning algorithms on collected images, including image classification, semantic segmentation and object detection.	82%
2.	Diabetic Retinopathy Diagnosis Through Computer-Aided Fundus Image Analysis	Jaskirat Kaur, Deepti Mittal & Ruchi Singla	2021	Computer-aided diagnostic assistance to an expert plays a vital part by aiding in the daily tasks of diagnosis of DR. As a result, numerous methods, such as morphology and thresholding, filtering, supervised methods, hybrid methods are being used to design such systems for the qualitative examination of retinal fundus images	97.38%



S.NO	PAPER	AUTHOR	YEAR	METHOD AND ALGORITHM	ACCURACY
3.	Deep Learning Techniques for Diabetic Retinopathy Classification	Mohammad Z. Atwany, Abdulwahab H. Sahyoun, Mohammad Yaqub	2022	Diabetic Retinopathy classification can be categorized to either binary classification which aims to detect the presence or absence of DR and multi-class classification, which determines the exact stage of DR. Consequently, Supervised, Self-supervised, and Transformer methods were developed to focus on lesion-based classification.	96.3%
4.	Deep learning architecture based on segmented fundus image features for classification of diabetic retinopathy	Sraddha Das, Krity Kharbanda, Suchetha M, Rajiv Raman, Edwin Dhas D	2021	They have used a convolution neural network (CNN) to train the classifier for performing classification. The CNN, constructed for classification, comprises a combination of squeeze and excitation and bottleneck layers, one for each class, and a convolution and pooling layer architecture for classification between the	98.7%

2.3 Problem Statement Definition:

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. Unfortunately, DR is not a reversible process, and treatment only sustains vision. Early detection and treatment of DR can significantly reduce the risk of vision loss.

Existing Diagnosis Methodology:
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.

Cause:

Diabetic retinopathy is a complication of diabetes, caused by high blood sugar levels damaging the back of the eye which may lead to loss of vision permanently.

Stages:

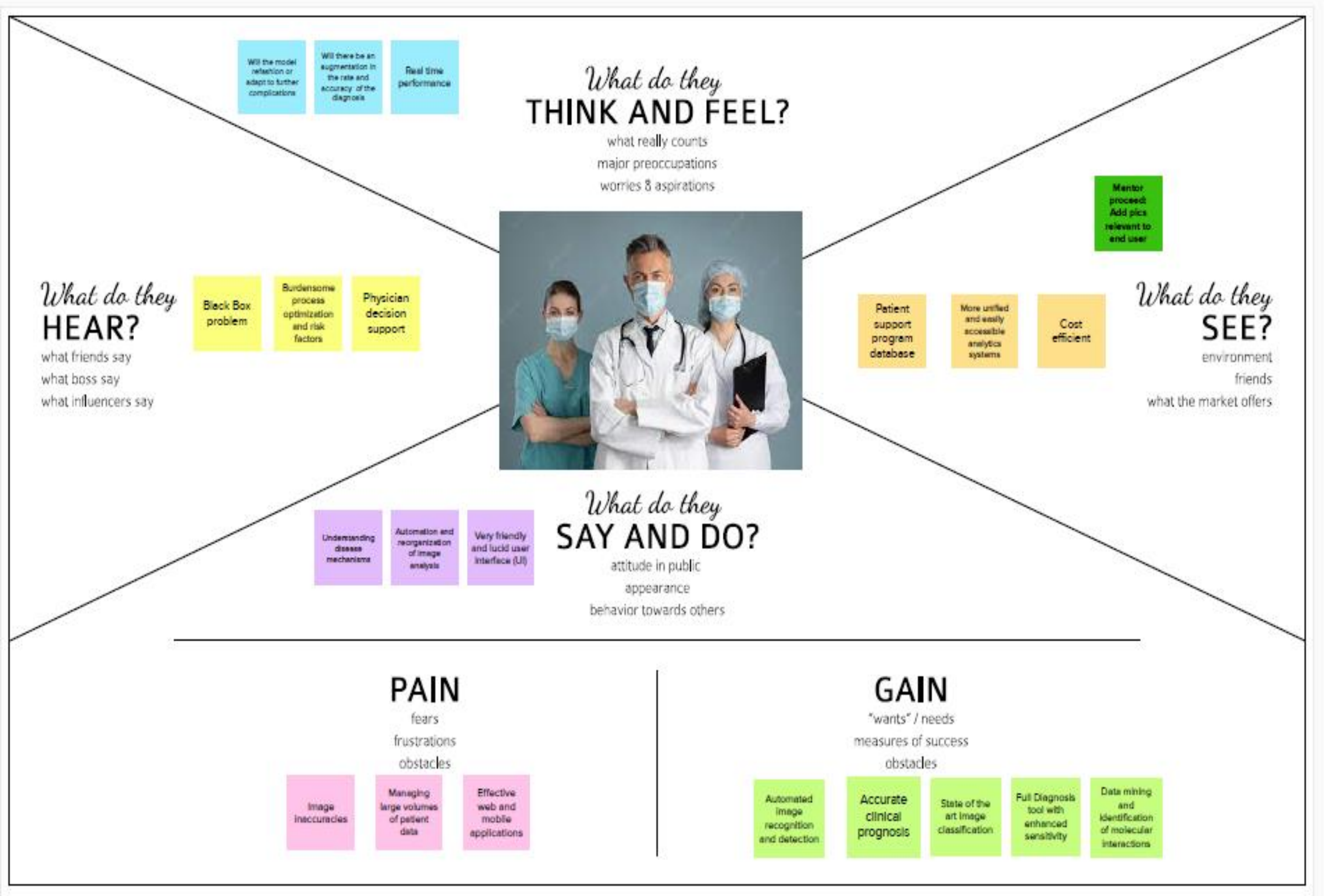
The early stages of DR don’t have any symptoms, the later stages may cause the vessels to leak small amount of blood into your retinas. Although it can be classified into mild, moderate, proliferative and non-proliferative.

Effects:

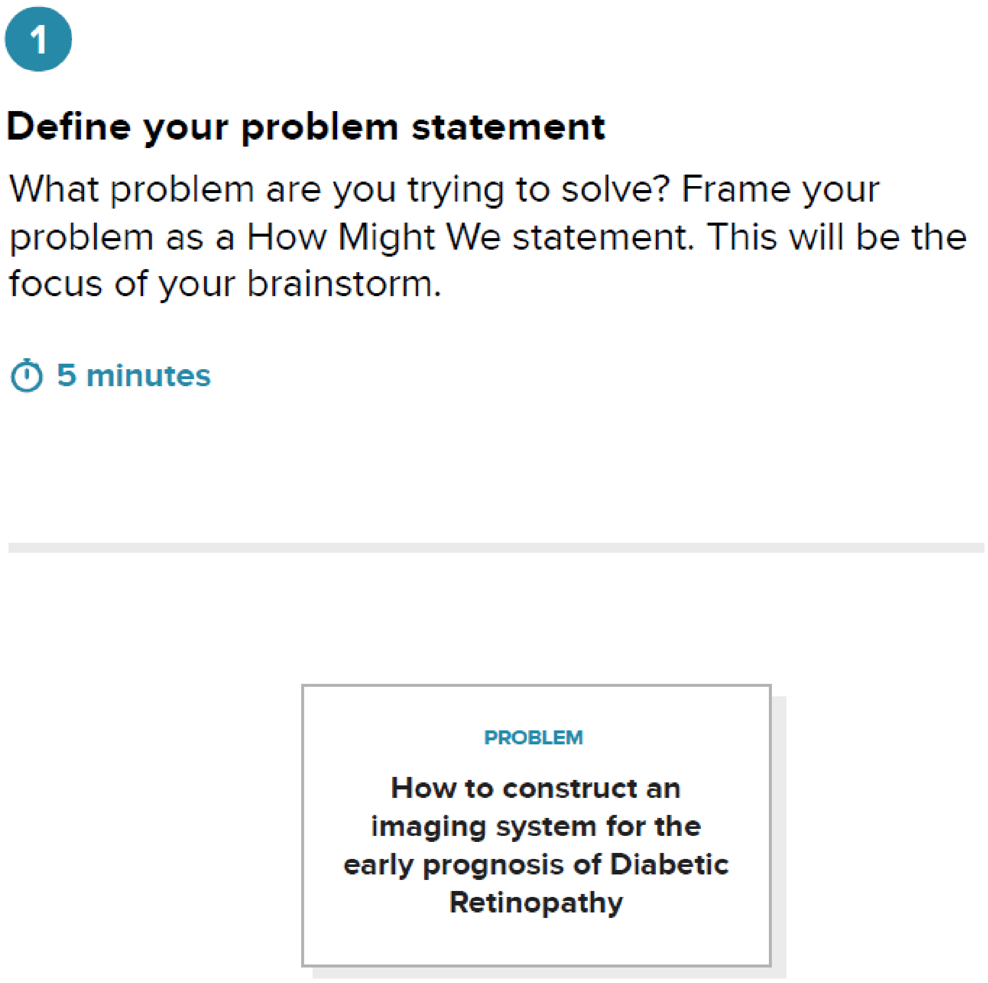
Vision disorder, blurred vision, distorted vision will occur. This problem occurs commonly for Diabetic patient.

3.Ideation Phase & Proposed Solution :

3.1 Empathy Map Canvas :


Will the model refashion or adapt to further complications
Will there be an augmentation in the rate and accuracy of the diagnosis
Real time performance
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Mentor proceed: Add pics relevent to end user
What do they
HEAR?
what friends say
what boss say
what influencers say
Black Box problem
Burdensome process optimization and risk factors
Physician decision support
Patient support program database
More unified and easily accessible analytics systems
Cost efficient
What do they
SEE?
environment
friends
what the market offers
Understanding disease mechanisms
Automation and reorganization of image analysis
Very friendly and lucid user interface (UI)
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Image inaccuracies
Managing large volumes of patient data
Effective web and mobile applications
GAIN
"wants" / needs
measures of success
obstacles
Automated image recognition and detection
Accurate clinical prognosis
State of the art image classification
Full Diagnosis tool with enhanced sensitivity
Data mining and identification of molecular interactions

3.2 Ideation and Brainstorming:



1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

⏱ 5 minutes

PROBLEM

How to construct an imaging system for the early prognosis of Diabetic Retinopathy

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

⏱ 10 minutes

Person 1

Identifying the age group that's mostly affected by diabetic retinopathy

How to advocate a personalized combination therapy

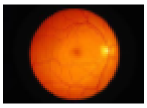
Classification of the disease into proliferative and non proliferative

Regular eye screening of type 2 diabetes patients showing symptoms

Person 2

Trying to make note of specific symptoms like neovascularization

initial fluorescein angiography



Study of preliminary retina images obtained via OCT

Integration of hardware and deep learning based software

Person 3

Regular blinking of eyes is advised to reduce eye discharge

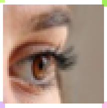
Intake nutritionist advised foods to control blood sugar and pressure

Regular and periodic usage of mild eye drops

Person 4

Glucose based positron emission tomography

Eye screening every once a year



Immediate medical advice in case of blurriness or patchiness in vision

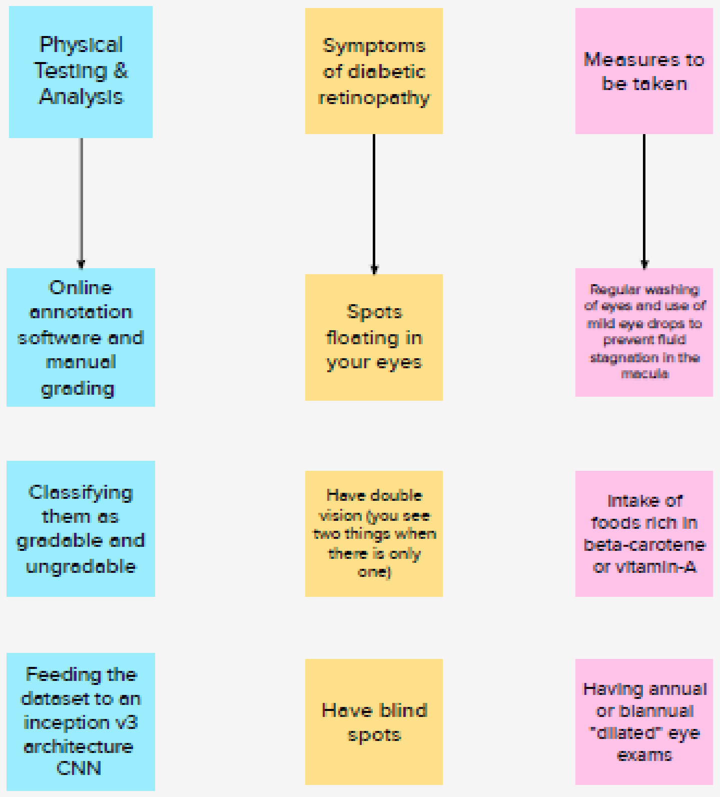
Preventive measures against macular edema caused due to diabetic retinopathy

3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

⏱ **20 minutes**



Physical Testing & Analysis

Online annotation software and manual grading

Classifying them as gradable and ungradable

Feeding the dataset to an inception v3 architecture CNN

Symptoms of diabetic retinopathy

Spots floating in your eyes

Have double vision (you see two things when there is only one)

Have blind spots

Measures to be taken

Regular washing of eyes and use of mild eye drops to prevent fluid stagnation in the macula

Intake of foods rich in beta-carotene or vitamin-A

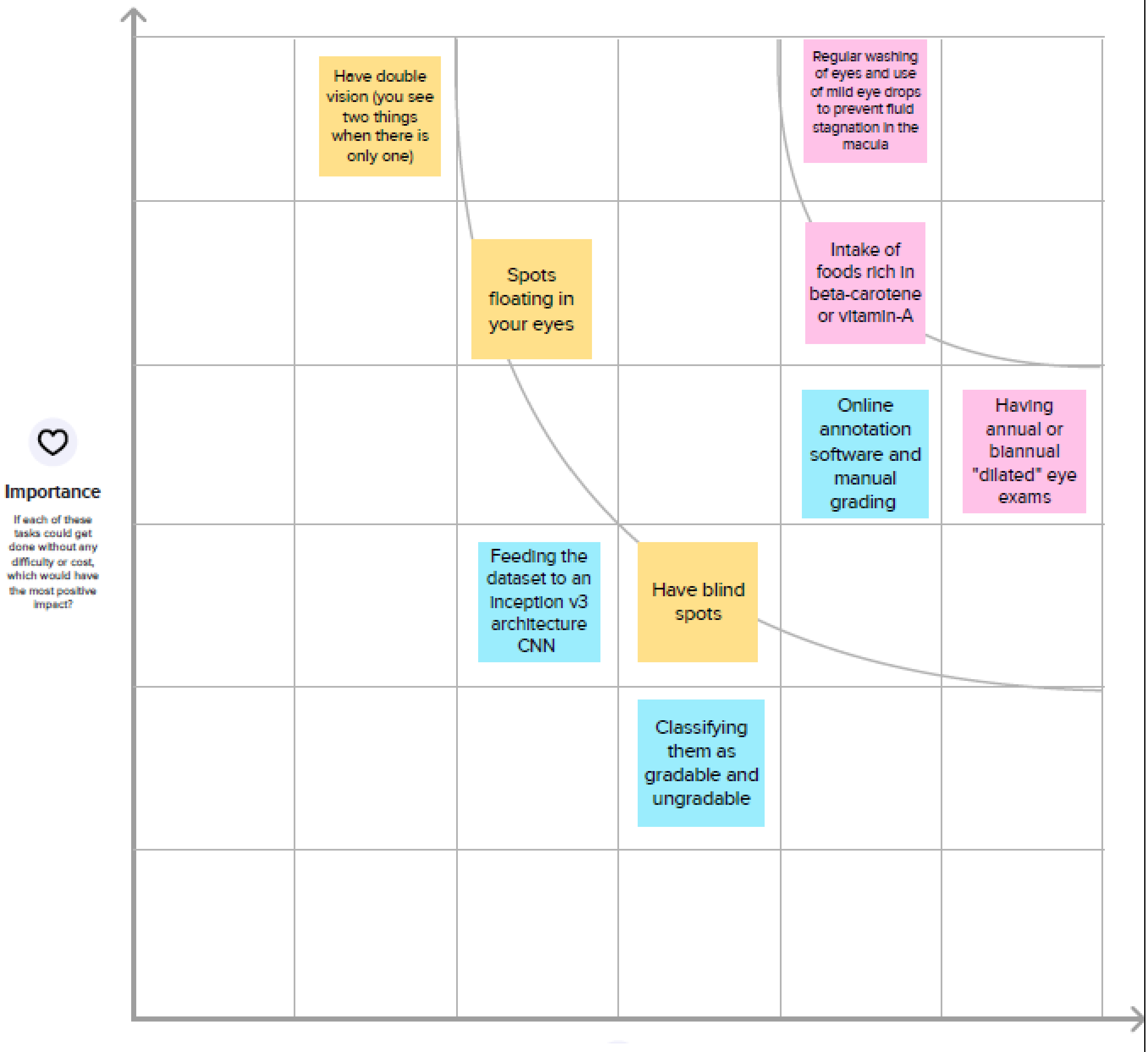
Having annual or biannual "dilated" eye exams

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes



Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Have double vision (you see two things when there is only one)

Regular washing of eyes and use of mild eye drops to prevent fluid stagnation in the macula

Spots floating in your eyes

Intake of foods rich in beta-carotene or vitamin-A

Online annotation software and manual grading

Having annual or biannual "dilated" eye exams

Feeding the dataset to an inception v3 architecture CNN

Have blind spots

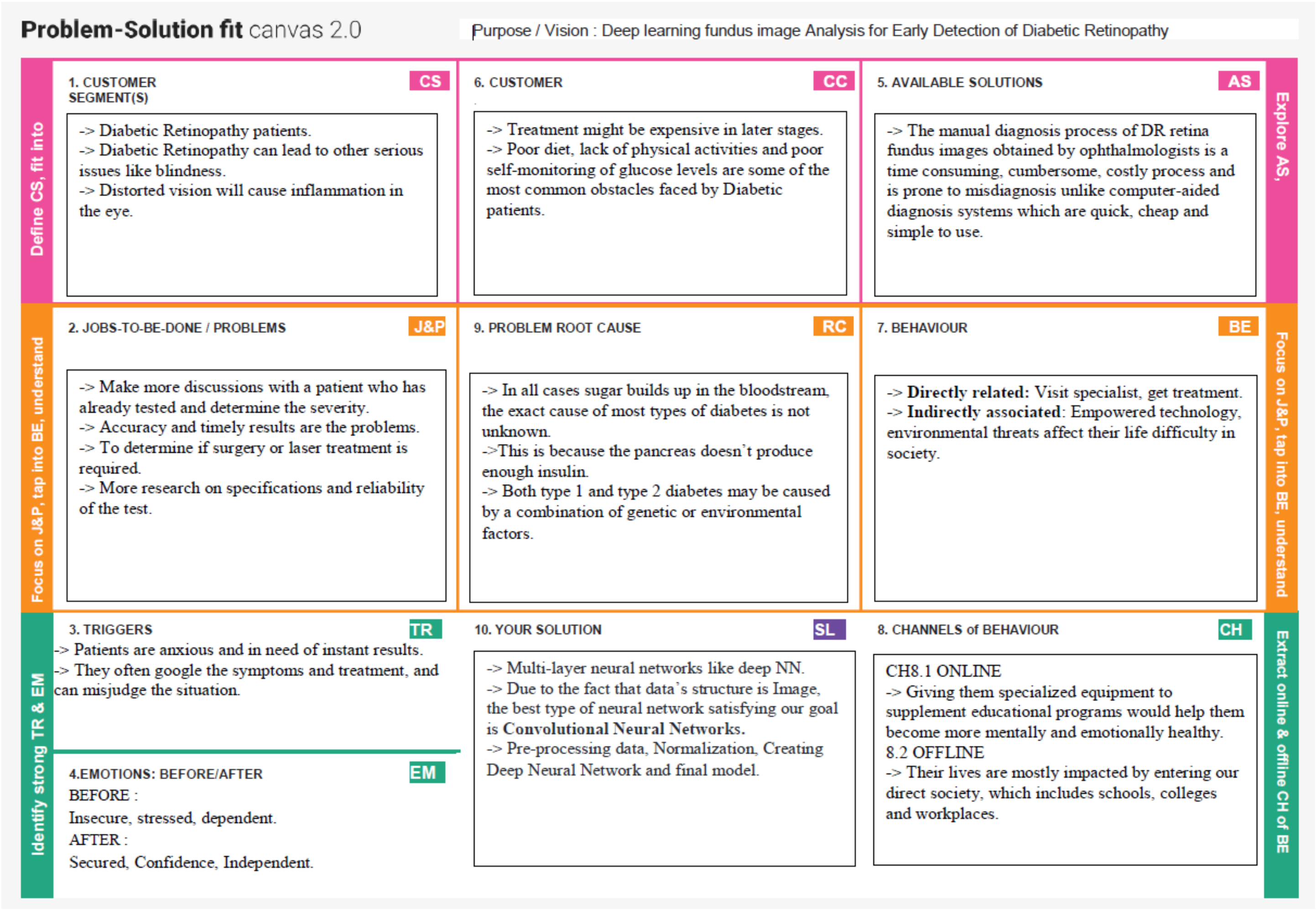
Classifying them as gradable and ungradable

3.3 :Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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 onlysustains vision. Early detection and treatment of DR can significantly reduce the risk of vision loss.
2.	Idea / Solution description	To automate the existing manual diagnosis of DR by using transfer learning-based image processing techniques to simplify, speed up thediagnosis and to improve the accuracy of the images acquired
3.	Novelty / Uniqueness	To develop a new CNN architecture based on renowned transfer learning models such as Inception v3, Resnet50 and Xception v3 etc. and accelerate the learning process. We also aim to increase the accuracy of the acquired images for better prognosis.
4.	Social Impact / Customer Satisfaction	This model will be discharged in the form of anapplication which embeds the CNN into a lucidUI. Therefore, the patients wouldn't have to undergo strenuous physical examination anymore. The application can further beextended in order to summarise the reports andconclusion of the diagnosis which would help the patient to acknowledge and understand the issue that he/she is suffering from (if any).
5.	Business Model (Revenue Model)	This can be very well classified under a B2C(Business to Consumer) model. The diagnostic capabilities of a hospital would increase exponentially and the app can be used effectively by physicians for the examination of diabetic as well as non-

		diabetic patients as and when they come for routine eye check-upsor screening etc.
6.	Scalability of the Solution	The proposed idea will result in the formulation of an adaptive CNN model which will automatically detect even the different types of DR (proliferative and non- proliferative). It will also be programmed to diagnose other eye related repercussions ofdiabetes such as glaucoma, macular edema and cataracts etc. Therefore, the proposed model can be concluded as a highly scalable one.

3.4 Proposed Solution Fit:



Problem-Solution fit canvas 2.0

Purpose / Vision : Deep learning fundus image Analysis for Early Detection of Diabetic Retinopathy

1. CUSTOMER SEGMENT(S) CS

-> Diabetic Retinopathy patients.
-> Diabetic Retinopathy can lead to other serious issues like blindness.
-> Distorted vision will cause inflammation in the eye.

6. CUSTOMER CC

-> Treatment might be expensive in later stages.
-> Poor diet, lack of physical activities and poor self-monitoring of glucose levels are some of the most common obstacles faced by Diabetic patients.

5. AVAILABLE SOLUTIONS AS

-> The manual diagnosis process of DR retina fundus images obtained by ophthalmologists is a time consuming, cumbersome, costly process and is prone to misdiagnosis unlike computer-aided diagnosis systems which are quick, cheap and simple to use.

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

-> Make more discussions with a patient who has already tested and determine the severity.
-> Accuracy and timely results are the problems.
-> To determine if surgery or laser treatment is required.
-> More research on specifications and reliability of the test.

9. PROBLEM ROOT CAUSE RC

-> In all cases sugar builds up in the bloodstream, the exact cause of most types of diabetes is not unknown.
->This is because the pancreas doesn't produce enough insulin.
-> Both type 1 and type 2 diabetes may be caused by a combination of genetic or environmental factors.

7. BEHAVIOUR BE

-> **Directly related:** Visit specialist, get treatment.
-> **Indirectly associated**: Empowered technology, environmental threats affect their life difficulty in society.

3. TRIGGERS TR

-> Patients are anxious and in need of instant results.
-> They often google the symptoms and treatment, and can misjudge the situation.

4.EMOTIONS: BEFORE/AFTER EM

BEFORE :
Insecure, stressed, dependent.
AFTER :
Secured, Confidence, Independent.

10. YOUR SOLUTION SL

-> Multi-layer neural networks like deep NN.
-> Due to the fact that data's structure is Image, the best type of neural network satisfying our goal is **Convolutional Neural Networks.**
-> Pre-processing data, Normalization, Creating Deep Neural Network and final model.

8. CHANNELS of BEHAVIOUR CH

CH8.1 ONLINE
-> Giving them specialized equipment to supplement educational programs would help them become more mentally and emotionally healthy.
8.2 OFFLINE
-> Their lives are mostly impacted by entering our direct society, which includes schools, colleges and workplaces.

Define CS, fit into
Explore AS,
Focus on J&P, tap into BE, understand
Focus on J&P, tap into BE, understand
Identify strong TR & EM
Extract online & offline CH of BE

4.Requirement Analysis:

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identifying and selecting dataset	The appropriate dataset to enhance the model's performance is the necessary to select.
FR-2	Training	It is required to import the libraries needed for the training of the model.

FR-3	Diagnosis	The training should ensure proper diagnosis and make sure to identify the true and false of the medical condition (DR).
FR-4	Analysis	Based on the training, the model should analyse the medical conditions (DR) in order to predict/detect the disease accurately.
FR-5	Testing	The trained model is tested with different data to ensure it has trained well to predict/detect the medical condition.
FR-6	Reporting	The result of the experiment gives the medical report of the disease (DR) so that the patient can understand the level of the disease.
FR-7	Treatment	The testing of the model gives us the level of the medical condition so that we can go for the required treatment.

Non-functional Requirements:

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

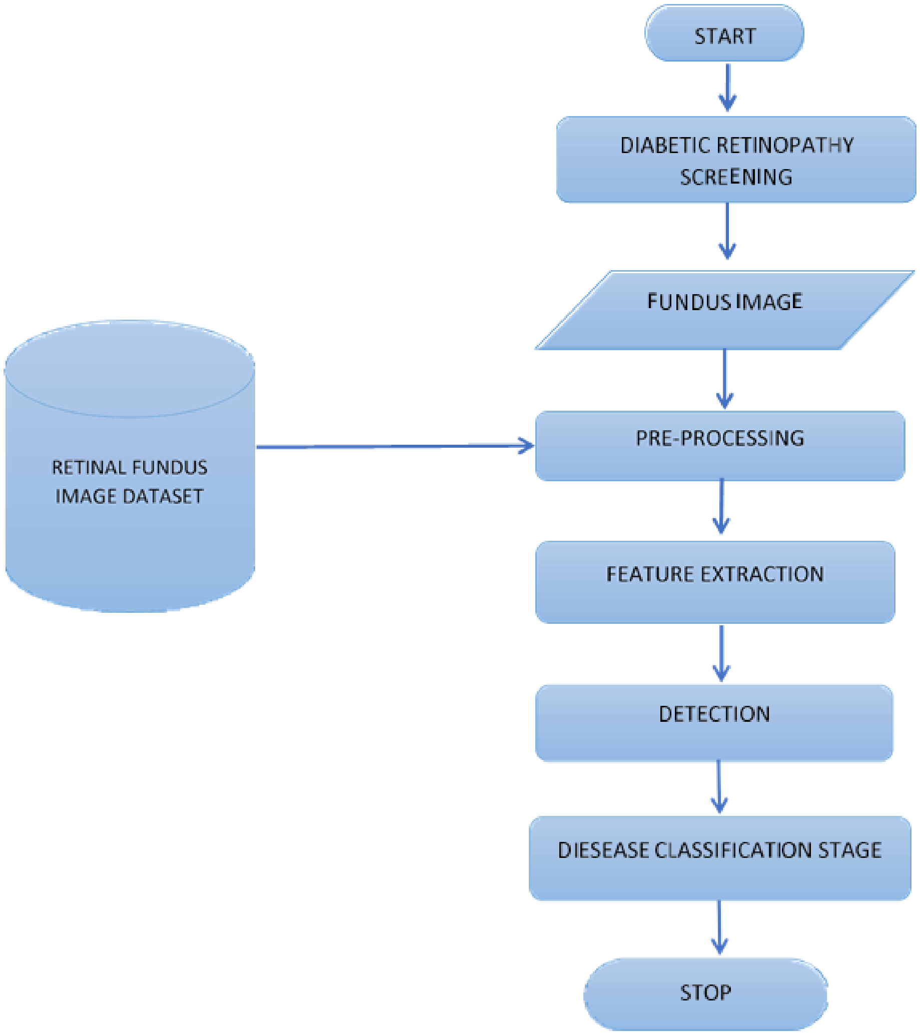
FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	User with basic understanding of the medical condition and computer knowledge can operate the system. Use friendly interface that can be accessed with ease by users.
NFR-2	**Reliability**	There is a chance of hardware failure or false positives when the testing data is more different than the training dataset. Permission is granted only by the administrator of the system.
NFR-3	**Performance**	If the system update fails or bugs in the code even though the system can roll back to its initial state. The performance of the model is meant to give speedy results for the patients.
NFR-4	**Availability**	The treatment should be available at low cost so that everyone with DR can find it beneficial.

NFR-5	**Scalability**	By processing more datasets for the reference of DR detection.

5.PROJECT DESIGN:-

5.1 DATA FLOW DIAGRAM:-

Data Flow Diagrams:


START
DIABETIC RETINOPATHY SCREENING
FUNDUS IMAGE
RETINAL FUNDUS IMAGE DATASET
PRE-PROCESSING
FEATURE EXTRACTION
DETECTION
DIESEASE CLASSIFICATION STAGE
STOP

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.

5.2 TECHNOLOGY ARCHITECTURE:-

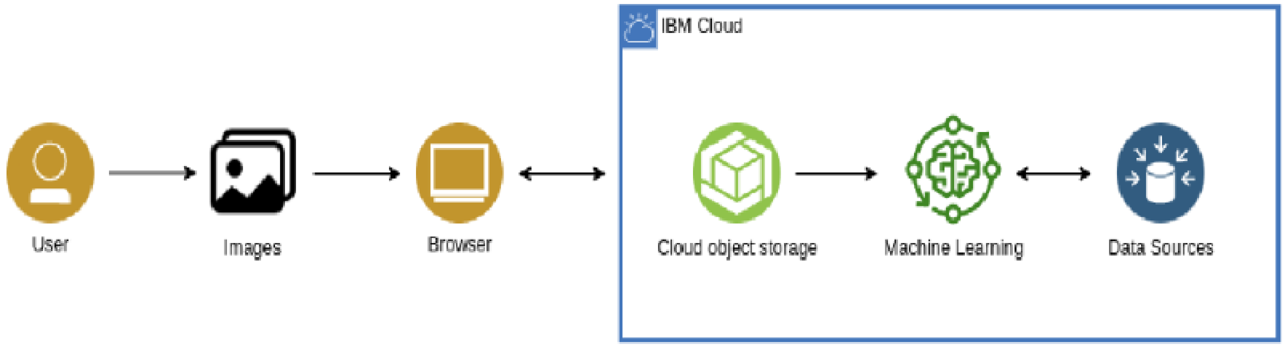


Table-1:Components& Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile app, Chatbot, etc.	HTML, CSS, JavaScript etc.
2.	Application Logic-1	Logic for a process in the application	Python, Flask
3.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.

4.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
5.	Machine Learning Model	Purpose of Machine Learning Model	Diabetic Retinopathy detection
6.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Cloud

Table-2:Application characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask, TensorFlow. Keras. Numpy, Pandas
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Built-in protection.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	3-tiers.
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers etc.)	Load balancer.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	It depends upon the input images.

5.3 USER STORIES:-

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile User)	Registration	USN-1	As a user, I can check whether I have retinopathy	I can upload or take image	High	Sprint-1

			or not by uploading the image of my eye by entering details			
	Login	USN-2	As a user, I can find the method more efficient and accurate.	It prevents the chances of unwanted infections In the patient’s eye	High	Sprint-1
		USN-3	As a user, I can find it portable and light weight		High	Sprint-2
	Physical feature	USN-4	. As a user, I can find it portable and light weight		Low	Sprint-2
	Safety	USN-5	As a user, I can be safe as the detection method is free from radiations.	Pain due to testing is the major fear factor that prevents patients from visiting the hospital	High	Sprint-4
Customer (Diabetic Patient)	Testing	USN-6	As a user, I can undergo testing without any fear of pain as this method is pain-free.	Pain due to testing is the major fear factor that prevents patients from visiting the hospital	Medium	Sprint-2
		USN-7	As a user, I will be comfortable as it requires minimum	The screenshot is carried out using a computer robot along	Low	Sprint-4

			human involvement	with aid of AI technology.		
	Results	USN-8	As a user, I can rely on the results without any suspicion.	The technique is almost 100% efficient as it involves Modern Techniques incorporated with Machine Learning.	High	Sprint-3
		USN-9	As a user, I can benefit from the result as it will help me know whether treatment is necessary or not.	It can prevent me from vision loss.	High	Sprint-1
		USN-10	As a user, I can get the results on the spot immediately after the screening process.	It prevents further delay in the treatment process.	Low	Sprint-4
Customer (Public sector/ Private sector)	Cost efficiency	USN-11	As a user, I can reach many people suffering from diabetes.	Diabetic patients are more vulnerable to DR	Medium	Sprint-1
		USN-12	As a user, I can create awareness among diabetic patients to	As the technique is of low cost, patients will	Low	Sprint-3

			undergo frequent screening.	find it useful.		
	Results	USN-13	As a user, I can complete the screening process within minutes for a single patient.	The random results generated by the device saves time.	High	Sprint-2

6.PROJECT PLANNING AND SCHEDULING:-

6.1 SPRINT PLANNING AND ESTIMATION:-

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story/ Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email or phone number and password, and confirming my password.	10	High	Priyanka s

Sprint -1	Dashboard	USN-2	As a user, I will Redirect to the dashboard after registration which shows the importance of DR.	10	Medium	Teena Amisha naazh And Swathika R
Sprint -2	Login	USN-3	As a user, I can log into the application by entering Login credentials.	5	High	Karpaga dharshini
Sprint -2	Upload Images	USN-4	As a user, I should be able to upload the image of eyeRetina.	10	High	Swathika R
Sprint2	Dashboard	USN-5	As a user, basedon my requirement I can navigate through the dashboard.	5	Medium	Priyanka S
Sprint3	Train the model	Task 1	As a developer, the dataset will be uploaded and trained by developed algorithm.	20	High	Teena Amisha Naazh
Sprint4	Testing & Evaluation	Task 2	As a developer, we tested the trained model using the	10	High	Swathika R

			provided dataset andmodel will be evaluated for accurate results.			
Sprint4	Display predicted result	USN-6	As a user, I can viewthe predicted resultin the dashboard.	10	High	Karpaga dharshini

Sprint	**Total story point**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date(Actual)**
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

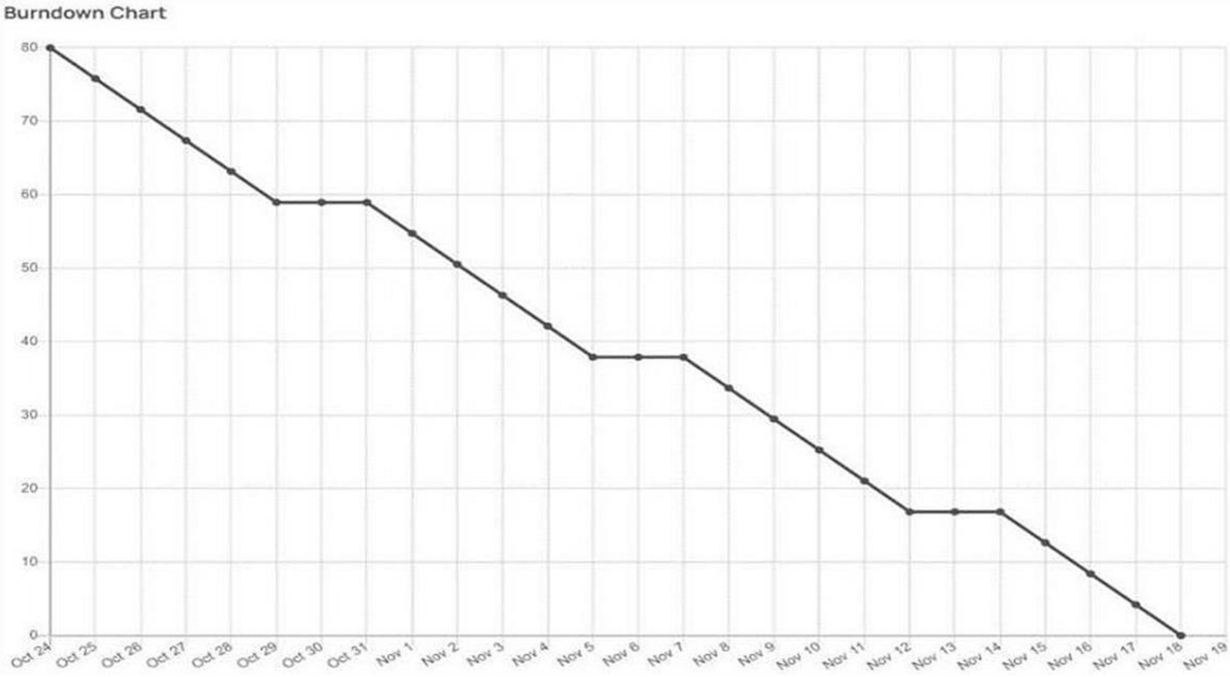
Velocity:

Imagine we have a 10-daysprint duration, and the velocityof the team is 20 (points per sprint). Let’s calculate the team’saverage velocity (AV)periteration unit (story points per day).

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

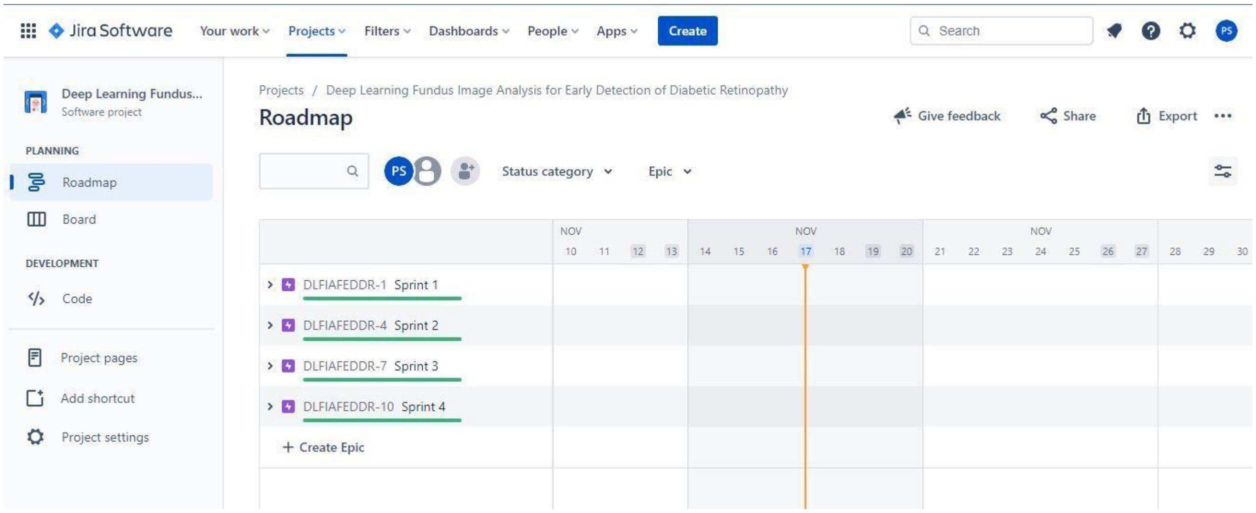
AV=20/6=3.33points per day.

6.2 Burn Down Chart & JIRA :



A burn down chart plots the amount of work remaining to perform against the amount of time. In agile software development approaches like Scrum, it is frequently employed. Burn down charts, however, can be used for any project that makes observable progress over time.

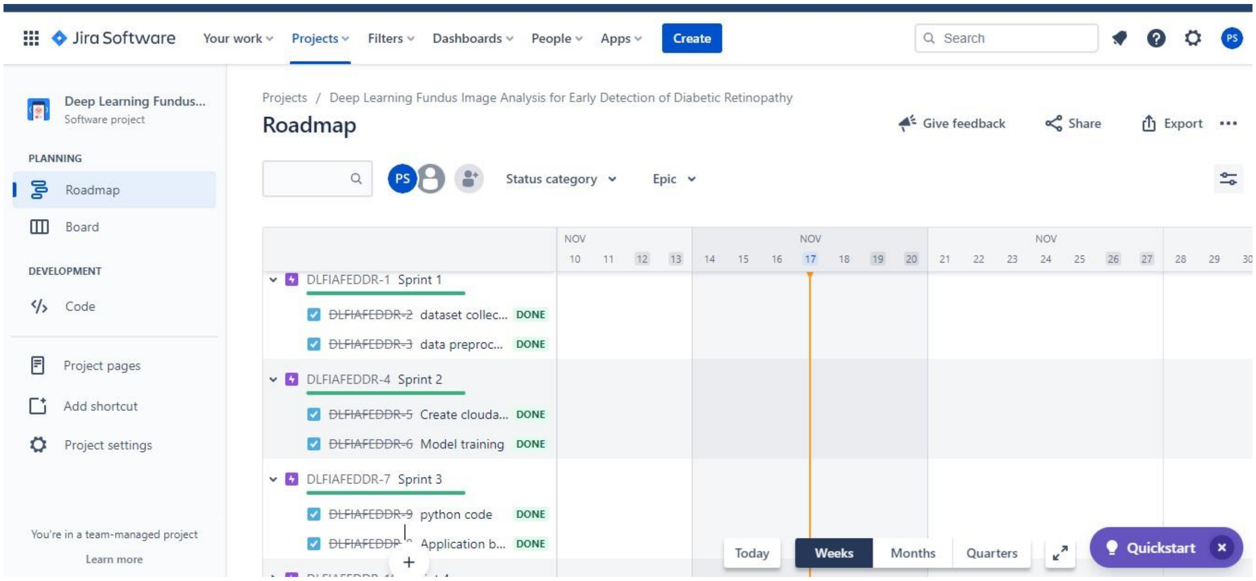
JIRA SCREENSHOTS:-



Projects / Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Roadmap

DLFIAFEDDR-1 Sprint 1
DLFIAFEDDR-4 Sprint 2
DLFIAFEDDR-7 Sprint 3
DLFIAFEDDR-10 Sprint 4
+ Create Epic



Projects / Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Roadmap

DLFIAFEDDR-1 Sprint 1
☑ ~~DLFIAFEDDR-2~~ dataset collec... DONE
☑ ~~DLFIAFEDDR-3~~ data preproc... DONE
DLFIAFEDDR-4 Sprint 2
☑ ~~DLFIAFEDDR-5~~ Create clouda... DONE
☑ ~~DLFIAFEDDR-6~~ Model training DONE
DLFIAFEDDR-7 Sprint 3
☑ ~~DLFIAFEDDR-9~~ python code DONE
☑ ~~DLFIAFEDDR-8~~ Application b... DONE

JIRA Folder is created to show the Scrum methodologies and Burn Down chart progress.

7.CODING AND SOLUTIONING:-

Feature 1:-

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

Feature 2:-

We have developed a multilayer deep convolutional neural network that classifies the user image of a eye to which expense 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 developed a messaging service for receiving message for the type of diabetics.

8.TESTING:-

8.1 TEST CASES:-

8.2 USER ACCEPTANCE TESTING:-

1. Purpose of Document:-

This document serves as a quick reference for the Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy project's test coverage and open issues as of the project's release for user acceptance testing.

2. **Defect Analysis:-** This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity4	Subtotal
By Design	5	4	2	3	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	15	30
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2

Won'tFix	0	5	2	1	8
Totals	17	14	13	21	65

3.Test-CaseAnalysis

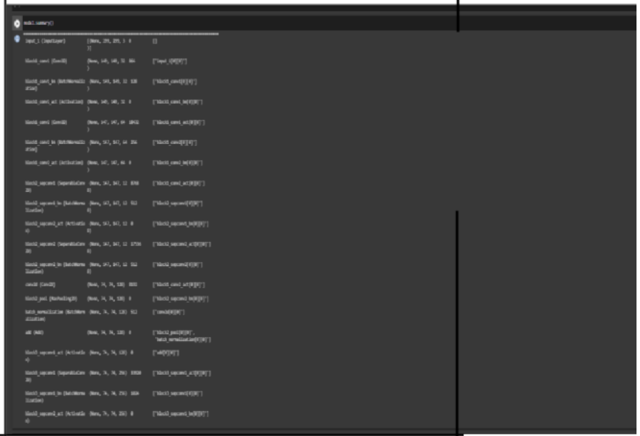
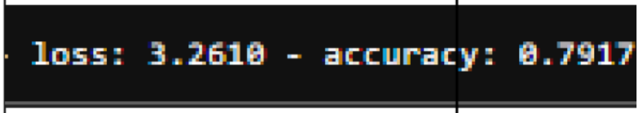
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	45	0	0	45
Security	2	0	0	2
Out-source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS:-

9.1 Performance Metrics:-

Model Performance Testing:

S. NO	Parameter	Values	Screenshot
1.	Model Summary	**Total params: 21,885,485** **Trainable params: 1,024,005** **Non-trainable params: 20,861,480**	 `model.summary()` [illegible]
2.	Accuracy	Training Accuracy – **0.7917** Validation Accuracy – **loss 3.2610**	 `- loss: 3.2610 - accuracy: 0.7917`
3.	Confidence Score(Only Yolo Projects)	Class Detected - Confide nce Score -	- - - - - - - -

10.ADVANTAGES AND DISADVANTAGES:-

10.1 ADVANTAGES:-

There are several advantages of using deep learning for fundus image analysis for early detection of diabetic retinopathy.

First, 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.

Second, deep learning is efficient at handling large amounts of data. This is important for medical image analysis, as medical images are often very large.

Third, deep learning 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.

Fourth, deep learning is able to learn from data with little supervision. This is important for medical image analysis, as often there is limited labelled data available.

Finally, deep learning 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.

10.2 DISADVANTAGES:-

There are several disadvantages of deep learning for early detection of diabetic retinopathy. One disadvantage is that 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.

11.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.

12.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.

13.APPENDIX:-

app.py:-

```
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 from flask import Flask, request,flash,
render_template, redirect,url_for from cloudant.client import
Cloudant from twilio.rest import Client
model = load_model(r"Updated-xception-diabetic-
retinopathy.h5") app = Flask(__name__)
app.secret_key="abc"
app.config['UPLOAD_FOLDER'] = "User_Images"
# Authenticate using an IAM API key

client = Cloudant.iam('08bcbaf0-260b-48e0-abdb-08db348afcf2-bluemix',
                    'yhZfUubpS3vS1vEKZSS37teD6IAUi8oLynOCQLIwnQsa', connect=True)
# Create a database using an initialized client
my_database = client.create_database('my_database') if
my_database.exists():     print("Database '{0}'
successfully created.".format('my_db'))
# default home page or route

user = ""

@app.route
('/') def
index():
    return render_template('index.html', pred="Login", vis ="visible")

@
app.route('/index'
) def home():
    return render_template("index.html", pred="Login", vis ="visible")
```

```
# registration page
@
app.route('/register',methods=["GET","POS
T"]) def register():      if request.method ==
"POST":
        name                               =
request.form.get("name")
mail                                       =
request.form.get("emailid")
mobile                                     =
request.form.get("num")
pswd = request.form.get("pass")
        data = {
'name': name,
'mail': mail,
            'mobile': mobile,
            'psw': pswd
        }
print(data
)
        query = {'mail': {'$eq': data['mail']}}
docs =
my_database.get_query_result(query)
print(docs)         print(len(docs.all()))
if (len(docs.all()) == 0):
            url = my_database.create_document(data)             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
")     else:
        return render_template('register.html')


@ app.route('/login',
methods=['GET','POST']) def login():
if request.method == "GET":         user
= request.args.get('mail')         passw =
request.args.get('pass')         print(user,
passw)
        query = {'mail': {'$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="")         else:             if ((user
== docs[0][0]['mail'] and passw == docs[0][0]['psw'])):                 flash("Logged in as " + str(user))
return render_template('index.html', pred="Logged in as "+str(user), vis ="hidden", vis2="visible")
else:
                return render_template('login.html', pred="The password is
wrong.")     else:
        return render_template('login.html')


@
app.route('/logout'
) def logout():
    return render_template('logout.html')
```

```
@app.route("/predict",methods=["GET",
"POST"]) def predict():      if request.method
== "POST":
        f = request.files['file']
        # getting the current path 1.e where app.py is present
basepath = os.path.dirname(__file__)
        #print ( " current path " , basepath )
        # from anywhere in the system we can give image but we
want that         filepath = os.path.join(str(basepath),
'User_Images', str(f.filename))         #print ( " upload folder is " ,
filepath )         f.save(filepath)         img =
image.load_img(filepath, target_size=(299, 299))         x =
image.img_to_array(img)  # ing 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)
index = [' No Diabetic Retinopathy ', ' Mild NPDR ',
                ' Moderate NPDR ', ' Severe NPDR ', ' Proliferative DR
']         result = str(index[prediction[0]])
        print(result)
        account_sid = 'AC8e0f2f5263d71c8f630a6486779cf08b'
auth_token = '30b489873afb3c47340070eabd6bfb15'

        client = Client(account_sid, auth_token)

        ''' Change the value of 'from' with the number
received from Twilio and the value of 'to'         with
the number in which you want to send message.'''
message = client.messages.create(
from_='+16075363206',                                   body
='Results: '+ result,                                   to
='+919445979800'
                              )
        return render_template('prediction.html', prediction=result, fname =
filepath)     else:
        return render_template("prediction.html")

if __name__ == "__main__":
    app.debug = True
app.run()
```

cloud.ipynb:-

```
from cloudant.client import Cloudant
client=Cloudant.iam('655489f8-18d0-4a44-a701-5de60570a973-
bluemix','Jc4eF6CXk72w0wGCsM_KUuXKVjsCcT4a54UKBXckK5Bv',connect=True)
my_database=client.create_database('my-database')
```

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, initial-scale=1.0"
/>     <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
    />
    <!-- JavaScript Bundle with Popper -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
    ></script>
    <style>
        #navbarRight {
margin-left: auto;
            padding-right:10px;
        }
        .navbar-brand{
padding-left:15px;
        }
    </style>
    <title>DR Predcition</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-light bg-
dark">         <div>
        <a class="navbar-brand" href="#" style="color:aliceblue">Diabetic Retinopathy Classification</a>
</div>
        {{msg}}
        <div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
          <ul class="navbar-nav mr-auto text-center" id="navbarRight">
            <li class="nav-item active">
              <a class="nav-link" href="index" style="color: aliceblue;">Home </a>
            </li>
            <li class="nav-item" style="visibility:{{ vis2 }}">
              <a class="nav-link" href="predict" style="color: aliceblue;">Prediction</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="login" style="color: aliceblue;">{{pred}}</a>
            </li>
            <li class="nav-item" style="visibility:{{ vis }}">
              <a class="nav-link" href="register" style="color: aliceblue;">Register</a>
            </li>
          </ul>
        </div>
      </nav>
      <br><br>
        <div class="jumbotron container">
          <h1 class="display-4">Diabetic Retinopathy</h1>
        <p class="lead">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.</p>          <hr class="my-4">
        <div class="d-flex justify-content-center">
          <img style="width:70vw;" src="static/diabetic-retinopathy-home.jpg">
          </div>
      </div>
  </body>
</html>
```

login.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, initial-scale=1.0"
/>     <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
    />
    <!-- JavaScript Bundle with Popper -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
    ></script>
    <style>
        #navbarRight          {
margin-left:              auto;
padding-right:10px;

        }
        .navbar-brand{
padding-left:15px;
        }
    </style>
    <title>DR Predcition</title>
  </head>
  <form action="",method='POST'>
    <nav class="navbar navbar-expand-lg navbar-light bg-
dark">         <div>
        <a class="navbar-brand" href="#" style="color:aliceblue">User Login</a>
</div>
        <div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
          <ul class="navbar-nav mr-auto text-center" id="navbarRight">
            <li class="nav-item active">
              <a class="nav-link" href="index" style="color: aliceblue;">Home </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="login" style="color: aliceblue;">Login</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="register"style="color: aliceblue;">Register</a>
            </li>
          </ul>
        </div>
      </nav>
      <br><br>
      <form class="form-inline" action="/login" method="GET">
      <div class="container" style="width: 600px; height: 600px;">
        <div class="mb-3 d-flex justify-content-center"><script src="https://cdn.lordicon.com/xdjxvujz.js"></script>
            <lord-icon
                src="https://cdn.lordicon.com/elkhjhci.json"
trigger="hover"
                style="width:200px;height:200px">
            </lord-icon></div>
            <div class="mb-3">
                <input type="email" class="form-control" id="exampleInputEmail1" name="mail" aria-
describedby="emailHelp" placeholder="Enter Registered Mail ID">
              </div>
              <div class="mb-3">
                <input type="password" class="form-control" id="exampleInputPassword1" name="pass" placeholder="Enter
Password">
              </div>
              <div class="mb-3">
              <button type="submit form-control" class="btn btn-dark btn-primary" style="width:100%;"
type="submit">Login</button>
            </div>
            {{pred}}
      </div>
    </form>
  </body>
</html>
```

logout.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, initial-scale=1.0"
/>     <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
    />
    <!-- JavaScript Bundle with Popper -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
    ></script>
    <style>
        #navbarRight {
margin-left: auto;
padding-right:10px;
        }
        .navbar-brand{
            padding-left:15px;
        }
    </style>
    <title>DR Predcition</title>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-light bg-
dark">         <div>
        <a class="navbar-brand" href="#" style="color:aliceblue">Diabetic Retinopathy</a>
</div>
        <div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
          <ul class="navbar-nav mr-auto text-center"
id="navbarRight">             <li class="nav-item active">
              <a class="nav-link" href="index" style="color: aliceblue;">Home </a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="login" style="color: aliceblue;">Login</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="register"style="color: aliceblue;">Register</a>
            </li>
          </ul>
        </div>
      </nav>
      <br><br>
      <div class="d-flex justify-content-center">
        <div class="row d-flex display-3 justify-content-center">
Successfully Logged Out!
            <br><br>
            <a href="login" class="btn btn-lg btn-dark">Login for more Information</a>
              </div>
        </div>
  </body>
</html>
```

prediction.html:-

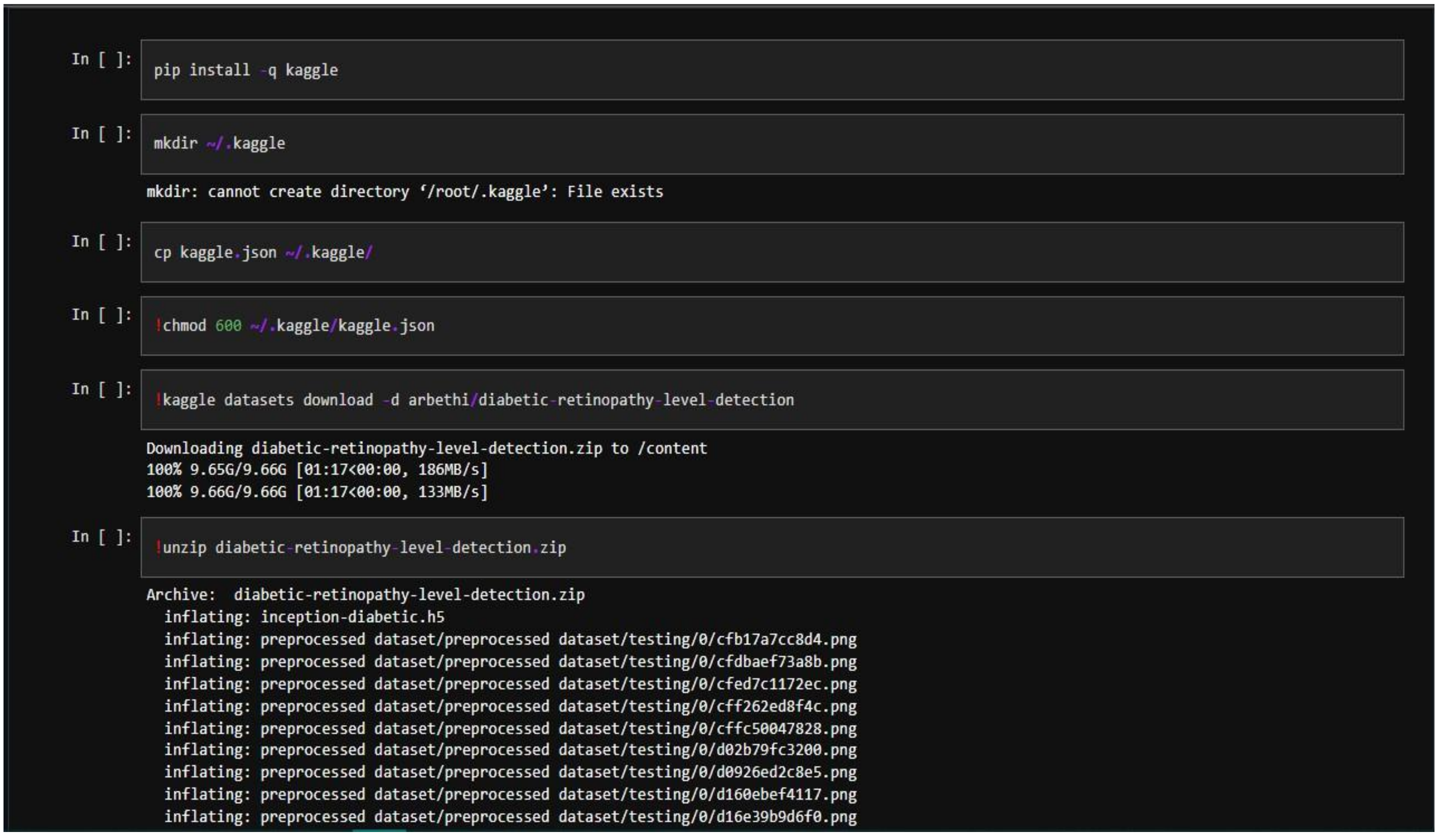
```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>   <!-- CSS only -->
  <link                    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"                                                                  integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous" />
  <!-- JavaScript Bundle with Popper -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
  <style>
    #navbarRight {
margin-left: auto;
padding-right: 10px;
    }

    .navbar-brand {
      padding-left: 15px;
    }

    .row {
width:
90%;
    }
  </style>
  <title>DR Predcition</title>
</head>

<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-dark">
    <div>
      <a class="navbar-brand" href="#" style="color:aliceblue">Diabetic Retinopathy Classification</a>
</div>
    <div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
      <ul class="navbar-nav mr-auto text-center" id="navbarRight">
        <li class="nav-item active">
          <a class="nav-link" href="index" style="color: aliceblue;">Home </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="logout" style="color: aliceblue;">Logout</a>
        </li>
      </ul>
    </div>
  </nav>
  <br><br>
  <div class="container justify-content-center" style="width:700px">
    <form action = "/predict" method = "POST" enctype="multipart/form-data">
    <label for="formFileLg" class="form-label">Upload Image</label>
    <input class="form-control form-control-lg" name ="file" type="file"
/>     <br>
    <button class="btn btn-lg btn-dark" type = "submit">Predict</button>
</form>
```

```
    <br>
    <h1>{{prediction}}</h1>
  </div>
  <br><br><br>
  <div class="d-flex justify-content-center" >
      <img src="static/level.png" style="width: 90%">
  </div>
</body>
</html>
```

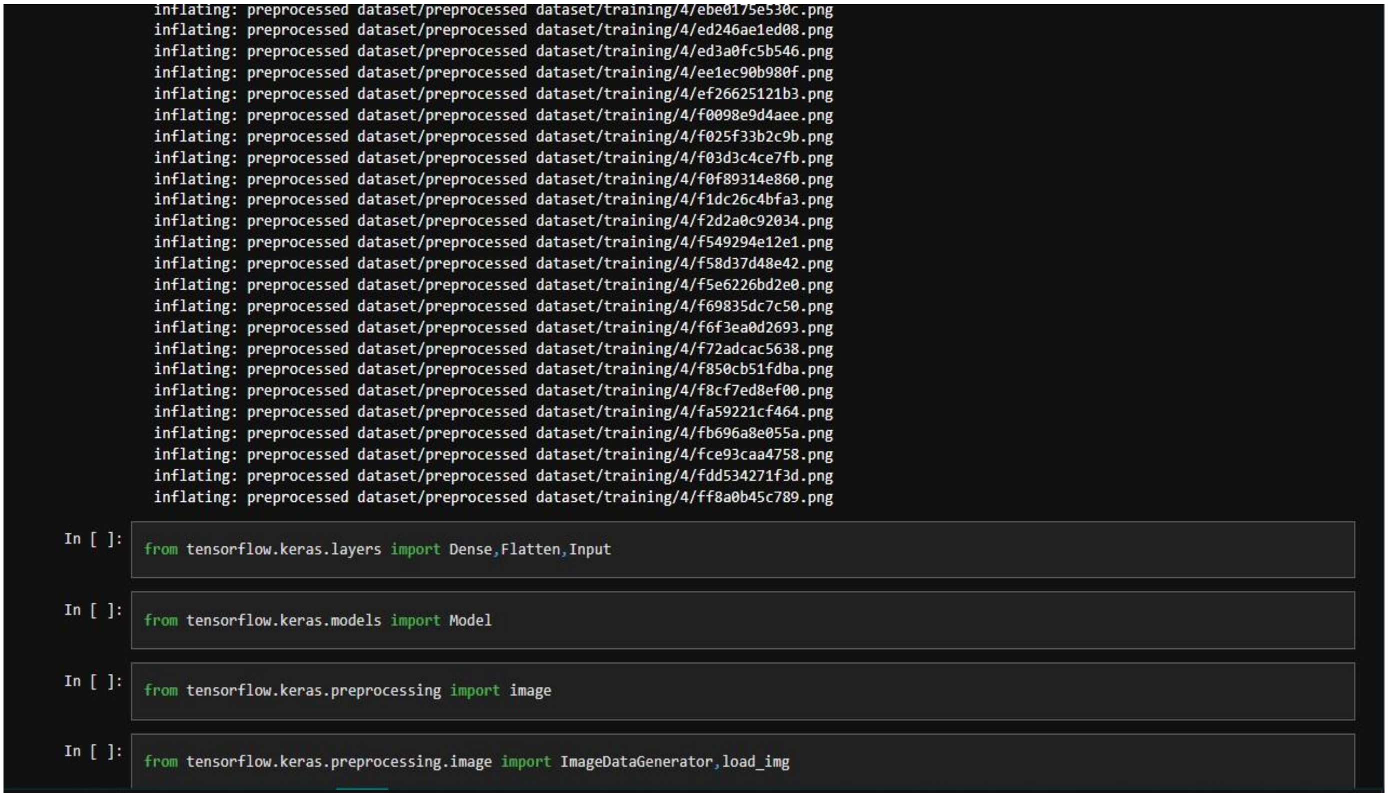
register.html:-

```
<!-- <!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, initial-scale=1.0"
/>     <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
      integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
    />
    <!-- JavaScript Bundle with Popper -->
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
    ></script>
    <style>
        #navbarRight {
margin-left: auto;
padding-right:10px;

        }
        .navbar-brand{
            padding-left:15px;
        }
    </style>
    <title>DR Predcition</title>
  </head>
  <form action="{{url_for('register')}}" method="post" >
    <nav class="navbar navbar-expand-lg navbar-light bg-
dark">         <div>
        <a class="navbar-brand" href="#" style="color:aliceblue">Registration</a>
</div>
        <div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
          <ul class="navbar-nav mr-auto text-center" id="navbarRight">
            <li class="nav-item active">
              <a class="nav-link" href="index" style="color: aliceblue;">Home </a>
            </li>
      <li class="nav-item">
       <a class="nav-link" href="login" style="color: aliceblue;">Login</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="register"style="color: aliceblue;">Register</a>
      </li>
     </ul>
    </div>
   </nav>
   <br><br>
   <form class="form-inline" method ="POST">
   <div class="container" style="width: 600px; height: 600px;">
    <div class="mb-3 d-flex justify-content-center"><script src="https://cdn.lordicon.com/xdjxvujz.js"></script>
      <lord-icon
        src="https://cdn.lordicon.com/elkhjhci.json"
trigger="hover"
        style="width:200px;height:200px">
      </lord-icon></div>
                      <div class="mb-3">
        <input type="text" class="form-control" id="exampleInputName" name = "name" aria-
describedby="nameHelp" placeholder="Enter Name">
       </div>
       <div class="mb-3">
        <input type="email" class="form-control" id="exampleInputEmail1" name="emailid" aria-
describedby="emailHelp" placeholder="Enter Mail ID">
       </div>
       <div class="mb-3">
        <input type="number" class="form-control" id="exampleInputNumber1" name="num"
ariadescribedby="numberHelp" placeholder="Enter Mobile number">
       </div>
       <div class="mb-3">
        <input type="password" class="form-control" id="exampleInputPassword1" name="pass" placeholder="Enter
Password">
       </div>
       <div class="mb-3">
       <button type="submit form-control" class="btn btn-dark btn-primary" style="width:100%;">Register</button>
      </div>
                      <div class="mb-3 d-flex justify-content-center">
                      <a href="login" class="nav-link"> Already Registered: Login Here</a>
   </div>
   {{pred}}
   </div>
  </form>
 </body>
</html> -->
```

Python Notebook screenshots:-



```
In [ ]: pip install -q kaggle

In [ ]: mkdir ~/.kaggle

        mkdir: cannot create directory ‘/root/.kaggle’: File exists

In [ ]: cp kaggle.json ~/.kaggle/

In [ ]: !chmod 600 ~/.kaggle/kaggle.json

In [ ]: !kaggle datasets download -d arbethi/diabetic-retinopathy-level-detection

        Downloading diabetic-retinopathy-level-detection.zip to /content
        100% 9.65G/9.66G [01:17<00:00, 186MB/s]
        100% 9.66G/9.66G [01:17<00:00, 133MB/s]

In [ ]: !unzip diabetic-retinopathy-level-detection.zip

        Archive:  diabetic-retinopathy-level-detection.zip
          inflating: inception-diabetic.h5
          inflating: preprocessed dataset/preprocessed dataset/testing/0/cfb17a7cc8d4.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/cfdbaef73a8b.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/cfed7c1172ec.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/cff262ed8f4c.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/cffc50047828.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/d02b79fc3200.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/d0926ed2c8e5.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/d160ebef4117.png
          inflating: preprocessed dataset/preprocessed dataset/testing/0/d16e39b9d6f0.png
```



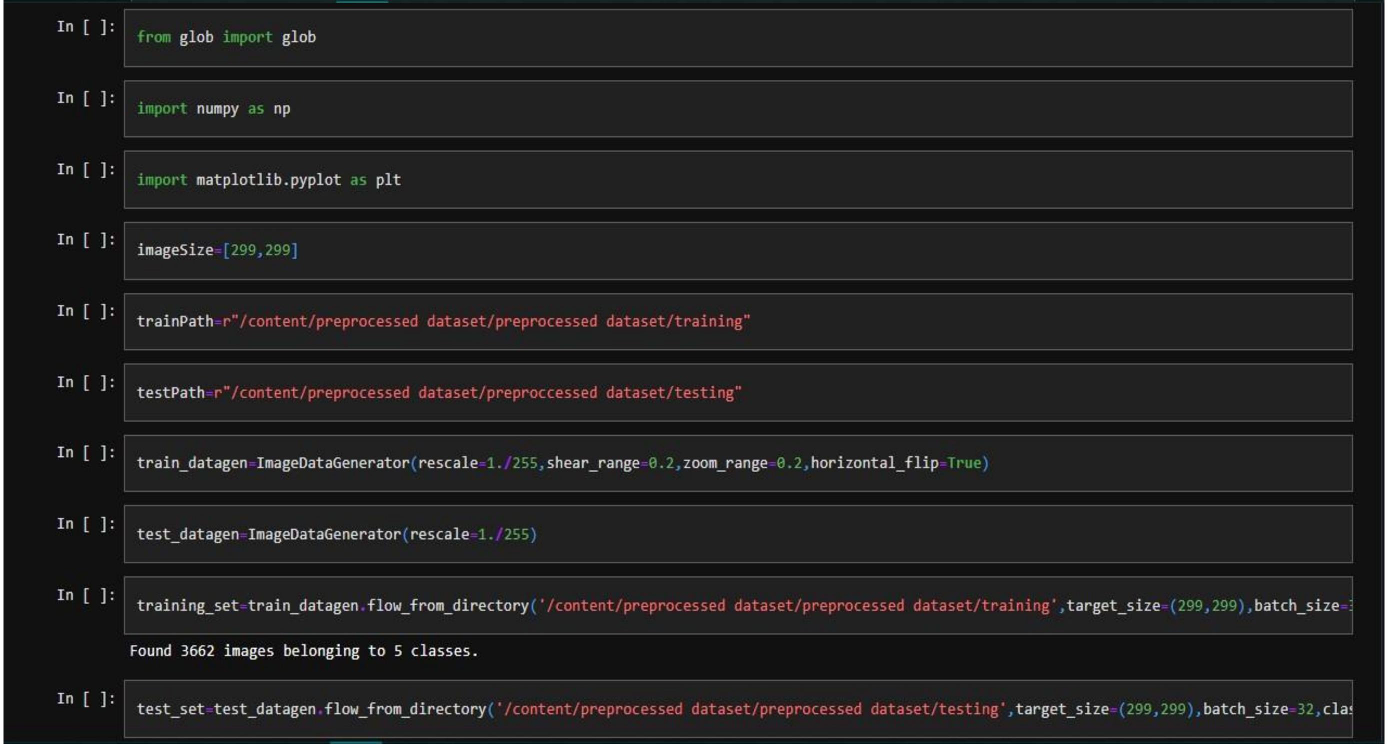
```
  inflating: preprocessed dataset/preprocessed dataset/training/4/ebe0175e530c.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/ed246ae1ed08.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/ed3a0fc5b546.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/ee1ec90b980f.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/ef26625121b3.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f0098e9d4aee.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f025f33b2c9b.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f03d3c4ce7fb.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f0f89314e860.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f1dc26c4bfa3.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f2d2a0c92034.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f549294e12e1.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f58d37d48e42.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f5e6226bd2e0.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f69835dc7c50.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f6f3ea0d2693.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f72adcac5638.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f850cb51fdba.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/f8cf7ed8ef00.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/fa59221cf464.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/fb696a8e055a.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/fce93caa4758.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/fdd534271f3d.png
  inflating: preprocessed dataset/preprocessed dataset/training/4/ff8a0b45c789.png
```

```
In [ ]: from tensorflow.keras.layers import Dense,Flatten,Input
```

```
In [ ]: from tensorflow.keras.models import Model
```

```
In [ ]: from tensorflow.keras.preprocessing import image
```

```
In [ ]: from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
```



```
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"
```

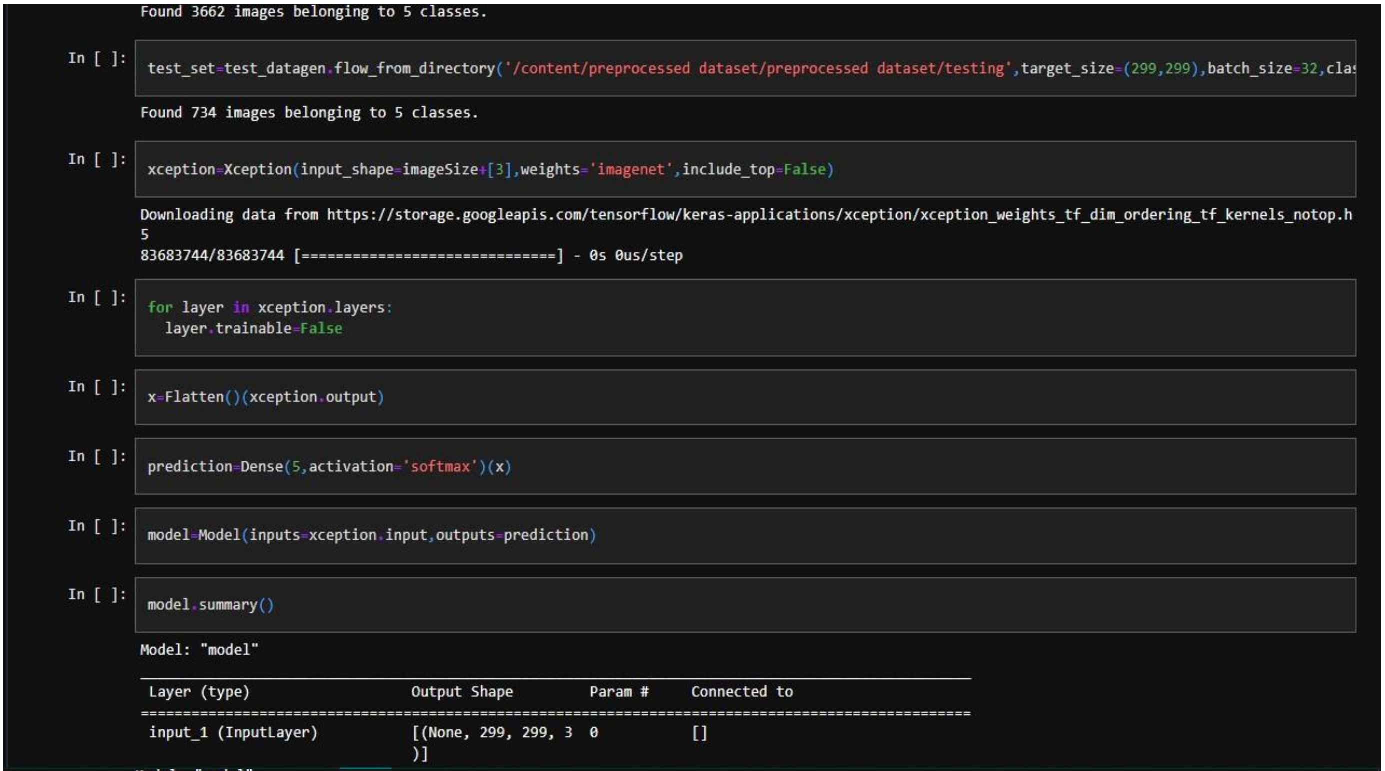
```
In [ ]: testPath=r"/content/preprocessed dataset/preproccessed dataset/testing"
```

```
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=3

        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,clas
```



```
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 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_notop.h
5
83683744/83683744 [==============================] - 0s 0us/step
```

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

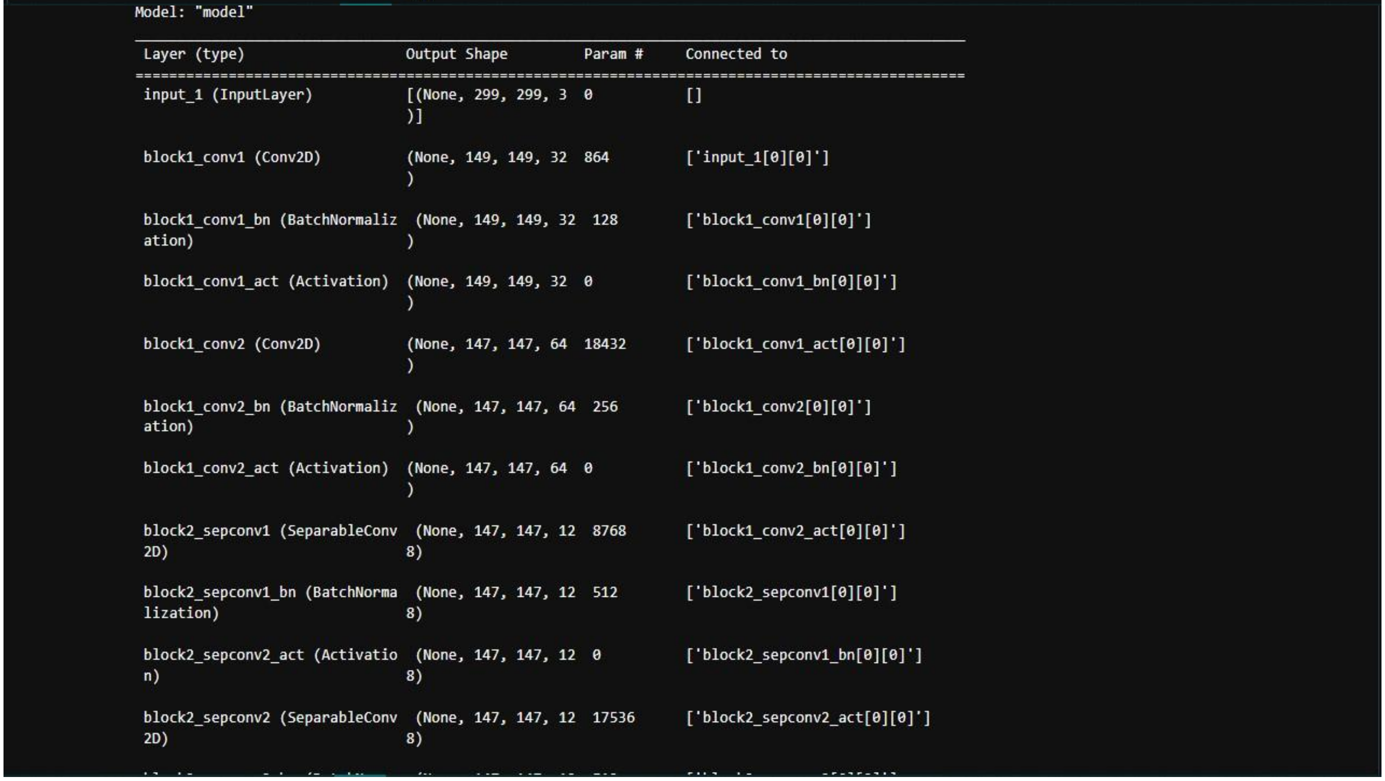
```
In [ ]: x=Flatten()(xception.output)
```

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

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

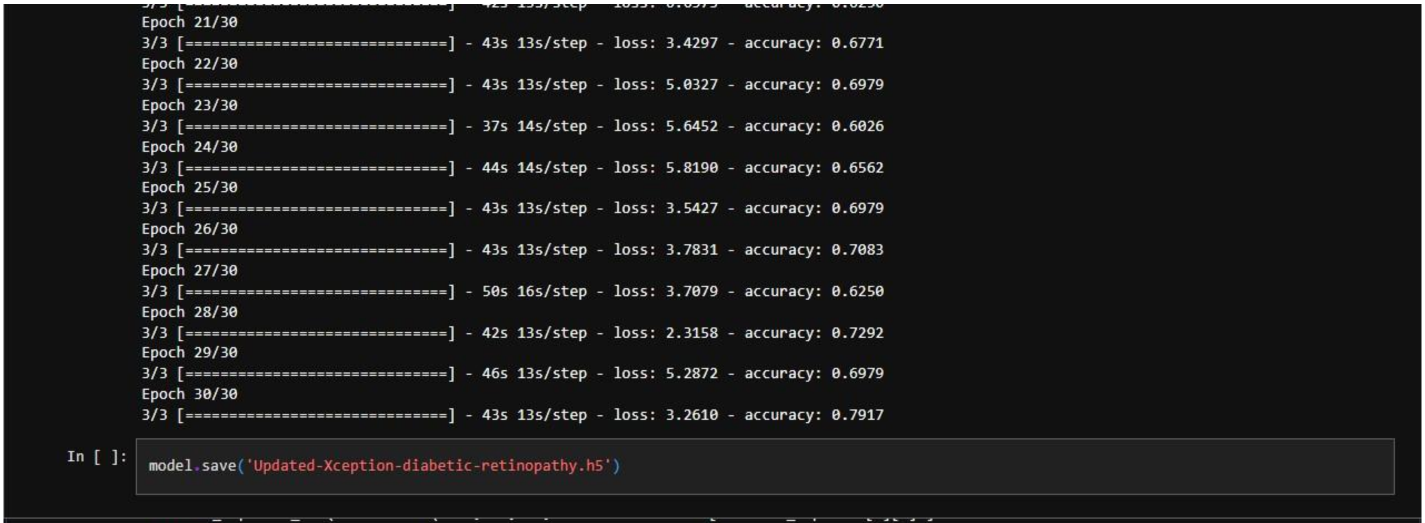
```
In [ ]: model.summary()
```

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



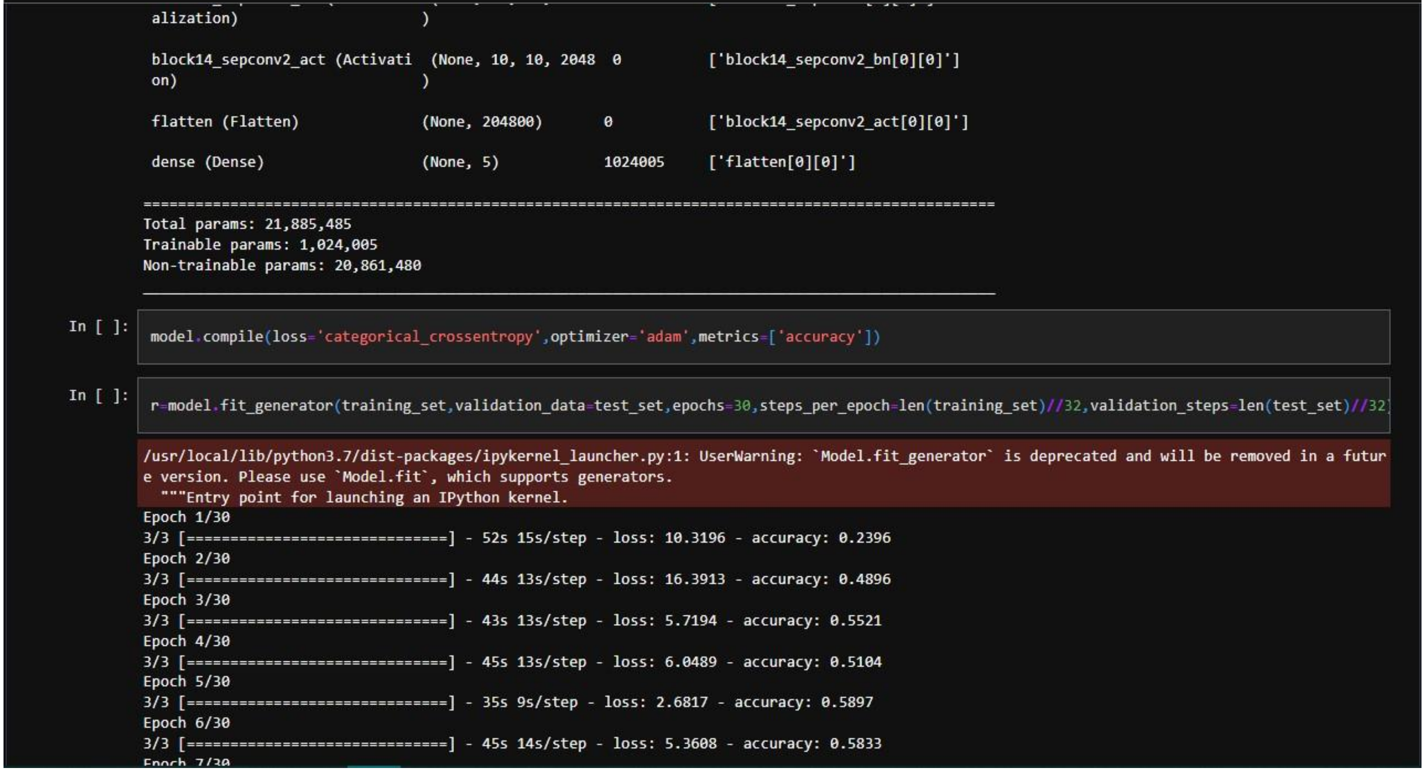
```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1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

```
 block1_conv1 (Conv2D)          (None, 149, 149, 32  864         ['input_1[0][0]']
                                )

 block1_conv1_bn (BatchNormaliz  (None, 149, 149, 32  128        ['block1_conv1[0][0]']
 ation)                         )

 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 (BatchNormaliz  (None, 147, 147, 64  256        ['block1_conv2[0][0]']
 ation)                         )

 block1_conv2_act (Activation)  (None, 147, 147, 64  0           ['block1_conv2_bn[0][0]']
                                )

 block2_sepconv1 (SeparableConv  (None, 147, 147, 12  8768       ['block1_conv2_act[0][0]']
 2D)                            8)

 block2_sepconv1_bn (BatchNorma  (None, 147, 147, 12  512        ['block2_sepconv1[0][0]']
 lization)                      8)

 block2_sepconv2_act (Activatio  (None, 147, 147, 12  0          ['block2_sepconv1_bn[0][0]']
 n)                             8)

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



```
Epoch 21/30
3/3 [==============================] - 43s 13s/step - loss: 3.4297 - accuracy: 0.6771
Epoch 22/30
3/3 [==============================] - 43s 13s/step - loss: 5.0327 - accuracy: 0.6979
Epoch 23/30
3/3 [==============================] - 37s 14s/step - loss: 5.6452 - accuracy: 0.6026
Epoch 24/30
3/3 [==============================] - 44s 14s/step - loss: 5.8190 - accuracy: 0.6562
Epoch 25/30
3/3 [==============================] - 43s 13s/step - loss: 3.5427 - accuracy: 0.6979
Epoch 26/30
3/3 [==============================] - 43s 13s/step - loss: 3.7831 - accuracy: 0.7083
Epoch 27/30
3/3 [==============================] - 50s 16s/step - loss: 3.7079 - accuracy: 0.6250
Epoch 28/30
3/3 [==============================] - 42s 13s/step - loss: 2.3158 - accuracy: 0.7292
Epoch 29/30
3/3 [==============================] - 46s 13s/step - loss: 5.2872 - accuracy: 0.6979
Epoch 30/30
3/3 [==============================] - 43s 13s/step - loss: 3.2610 - accuracy: 0.7917
```

```
In [ ]: model.save('Updated-Xception-diabetic-retinopathy.h5')
```



```
 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
__________________________________________________________________________________________________
```

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

```
In [ ]: 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
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a futur
e version. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/30
3/3 [==============================] - 52s 15s/step - loss: 10.3196 - accuracy: 0.2396
Epoch 2/30
3/3 [==============================] - 44s 13s/step - loss: 16.3913 - accuracy: 0.4896
Epoch 3/30
3/3 [==============================] - 43s 13s/step - loss: 5.7194 - accuracy: 0.5521
Epoch 4/30
3/3 [==============================] - 45s 13s/step - loss: 6.0489 - accuracy: 0.5104
Epoch 5/30
3/3 [==============================] - 35s 9s/step - loss: 2.6817 - accuracy: 0.5897
Epoch 6/30
3/3 [==============================] - 45s 14s/step - loss: 5.3608 - accuracy: 0.5833
Epoch 7/30
```

GITHUB LINK:- https://github.com/IBM-EPBL/IBM-Project-3977-1658677810.git

DEMO LINK:- https://drive.google.com/file/d/15-BMJbf8KdT3lU-fEs3RpqoqEqsurXVB/view?usp=drivesdk