PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

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 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.

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.

1.2 Purpose

Early detection (‘screening’) and timely treatment have been shown to prevent visual loss and blindness in patients with retinal complications of diabetes.In the next decade, projections for the United States are that the average age will increase, the number of people with diabetes in each age category will increase, and there will be an undersupply of qualified eye care providers, at least in the near-term. This “perfect storm” of healthcare trends will challenge the public health capacity to care for both patients with DR and people with diabetes at risk for this complication. If the previous scenario plays out, it will be necessary to either screen (perform early detection on) large numbers of people with diabetes for DR, ration access to eyecare, or both.

2. LITERATURE SURVEY

2.1 Existing problem

Diabetic retinopathy can cause abnormal blood vessels to grow out of the retina and block fluid from draining out of the eye. This causes a type of glaucoma (a group of eye diseases that can cause vision loss and blindness).

2.2 References

[1]. M. Chetoui, M. A. Akhloufi and M. Kardouchi, "Diabetic Retinopathy Detection Using Machine Learning and Texture Features", 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), pp. 1-4, 2018.

[2]. Kangrok Oh, Hae Min Kang, Dawoon Leem, Hyungyu Lee, Kyoung Yul Seo, Sangchul Yoon, "Early detection of diabetic retinopathy based on deep learning and ultra-wide-field fundus images", Scientific Reports 11, Article No:1897 1-9,2021.

[3]. Sheikh Muhammad Saiful Islam, Md Mahedi Hasan, Sohaib Abdullah, "Deep learning based early detection and grading of diabetic retinopathy using retinal fundus images" , arXiv preprint arXiv:1812.10595, 2018.

[4]. E. V. Carrera, A. González and R. Carrera, "Automated detection of diabetic retinopathy using SVM", 2017 IEEE XXIV International Conference on Electronics Electrical Engineering and Computing (INTERCON), pp. 1-4, 2017.

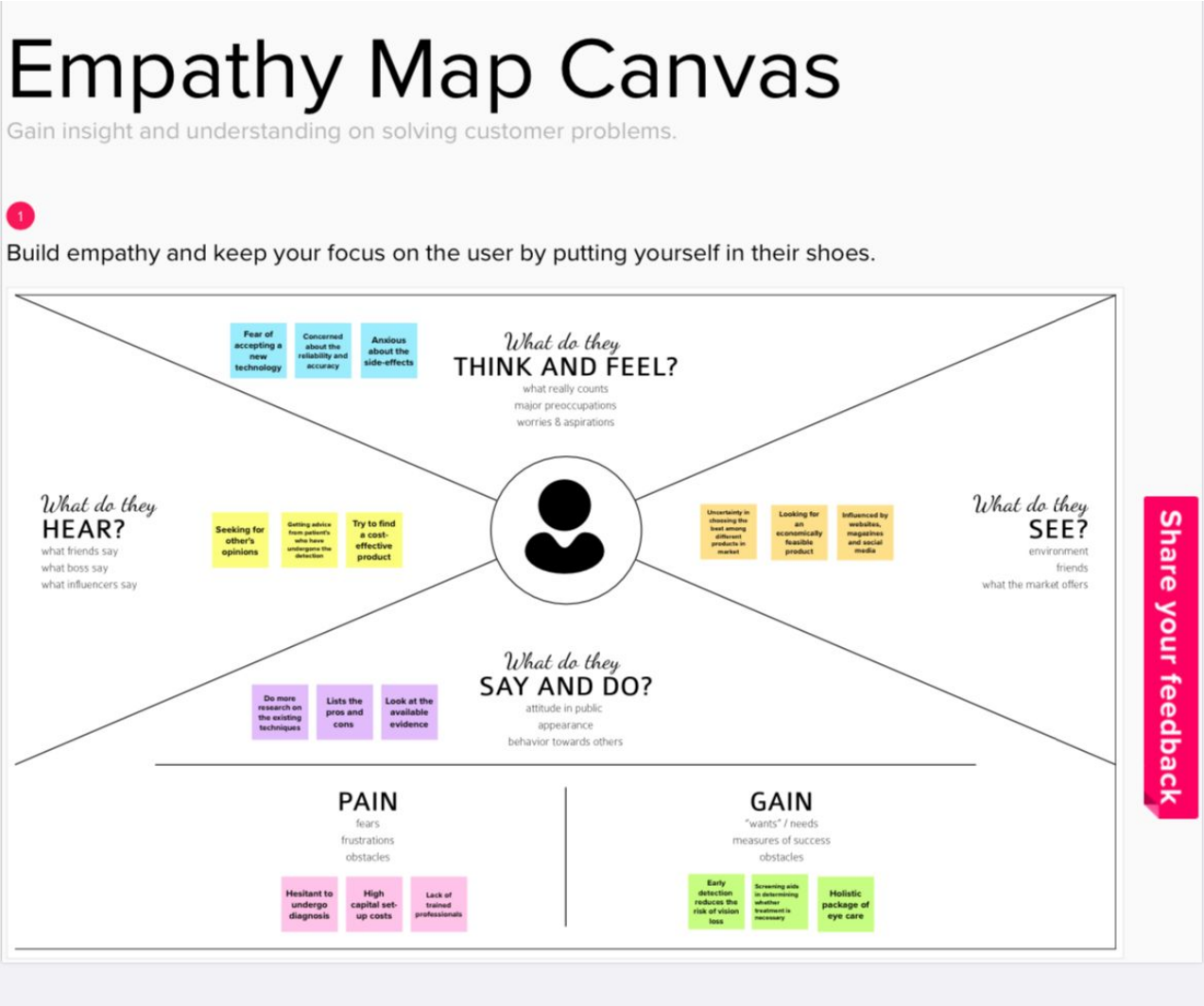
2.3 Problem Statement Definition

Diabetic retinopathy is caused by damage to the blood vessels in the tissue at the back of the eye (retina). Poorly controlled blood sugar is a risk factor. Early symptoms include floaters, blurriness, dark areas of vision and difficulty perceiving colours.If not detected and treated properly, it can lead to severe complications and loss of vision.Our project aims to

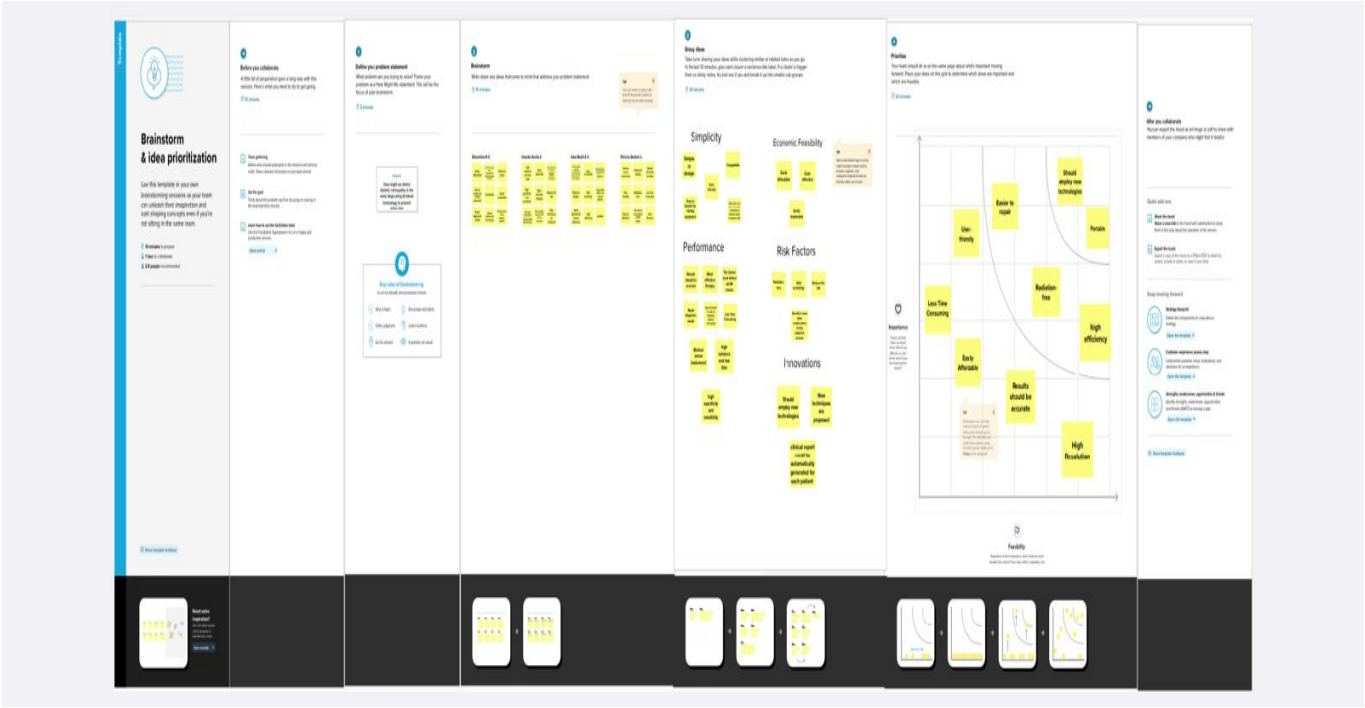
develop a better screening using techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



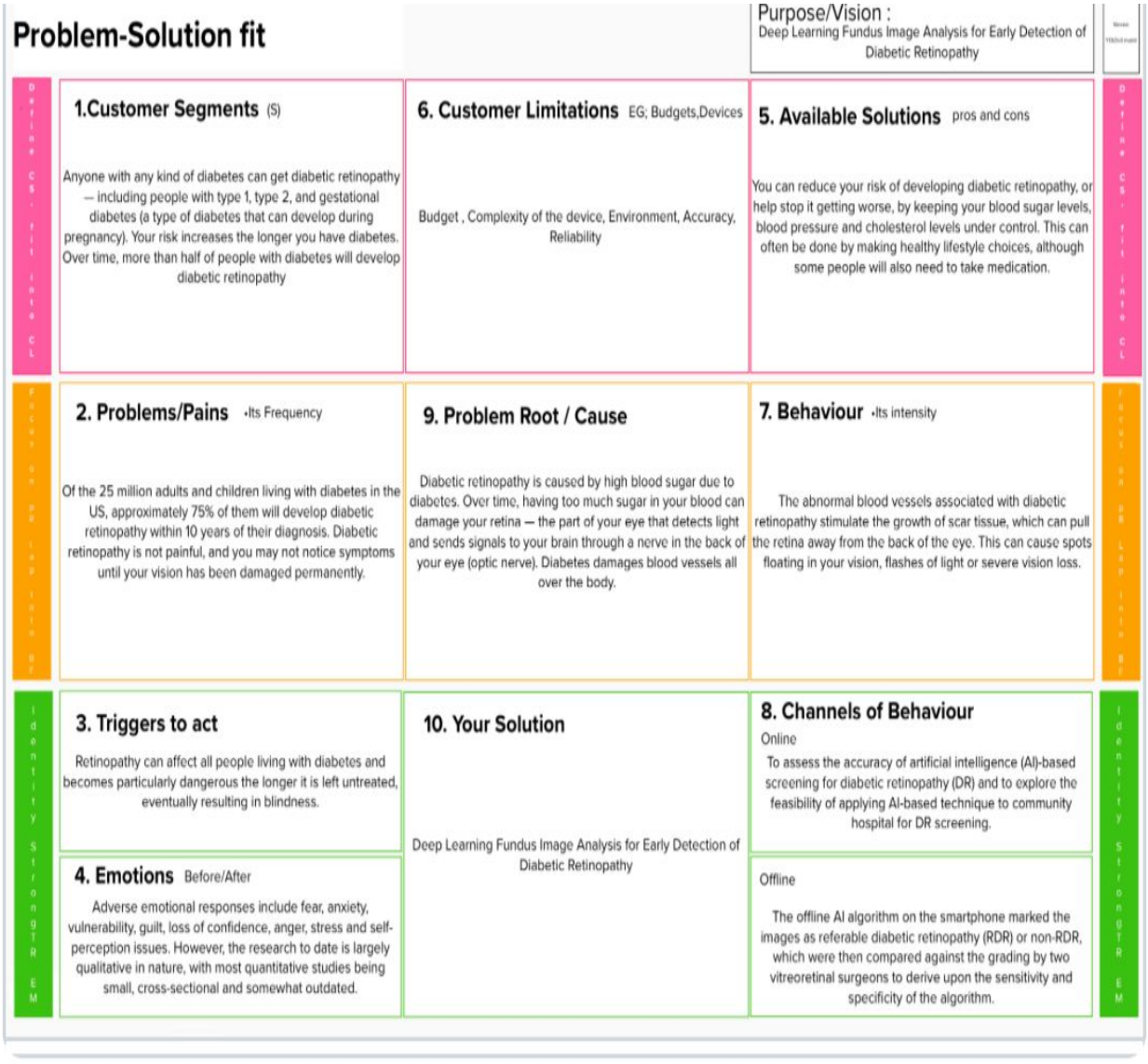
3.3 Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To detect Diabetic Retinopathy in early stage using AI based technology to prevent vision loss.
2.	Idea / Solution description	To detect all the stages of Diabetic Retinopathy efficiently in a cost-effective manner using emerging AI technologies.
3.	Novelty / Uniqueness	Minimal human involvement and radiation free detection method.
4.	Social Impact / Customer Satisfaction	Economically affordable and to create a society without vision loss due to Diabetic Retinopathy.
5.	Business Model (Revenue Model)	Easily marketable and profitable with greater customer satisfaction.
6.	Scalability of the Solution	Best design practices using right tools and framework to increase the throughput. Best user experience using realistic screening methods.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identifying the population eligible for screening	Determine the group to be screened based on best evidence and use registers to make sure people's details are collected and up to date
FR-2	Invitation and information	Invite the full cohort for screening, supplying information tailored appropriately for different groups to enable informed choice to participate
FR-3	Testing	Conduct screening test(s) using agreed/ recommended methods
FR-4	Referral of screen positives and reporting of screen-negative results	Refer all screen-positive results to appropriate services and make sure screen negatives are reported to individuals and they stay in the screening programme
FR-5	Diagnosis	Diagnose true cases and identify false positives
FR-6	Intervention/treatment/follow up	Intervene/treat cases appropriately; in some conditions, surveillance or follow up will also be required
FR-7	Reporting of outcomes	Collect, analyse and report on outcomes to identify false negatives and improve effectiveness and cost-effectiveness of screening programme

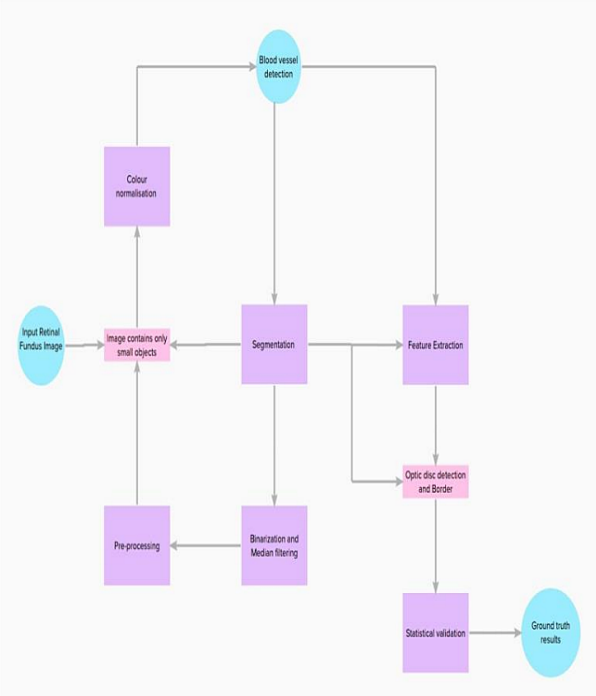
4.2 Non-Functional requirements

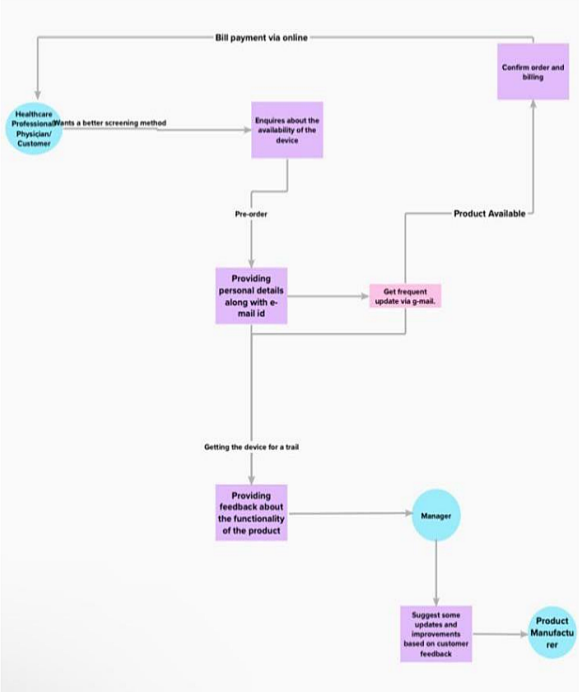
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Provides novel results for five different screening and clinical grading systems for diabetic retinopathy including state-of-the-art results for accurately classifying images according to clinical five-grade diabetic retinopathy.
NFR-2	**Security**	Deep Learning using AI can be more precise around sensitive organs and tissues, reduce blood loss, risk of infection, and pain during detection/screening.
NFR-3	**Reliability**	The ability of Deep Learning is to perform pattern recognition by creating complex relationships based

		on input data and then comparing it with performance standards is a big step.
NFR-4	**Performance**	AI in simple words means to accomplish a task mainly by a computer or a robot, with minimal involvement of human beings. Standard templates for drawing findings of the retina may improve accuracy of recording of results.
NFR-5	**Availability**	Healthcare affordability, quality, and accessibility can be amplified using this technology.
NFR-6	**Scalability**	It is possible to build on existing systems and take a stepwise approach to improving the effectiveness of current approaches so that high-quality systematic diabetic retinopathy screening becomes a universal offer to all people with diabetes.

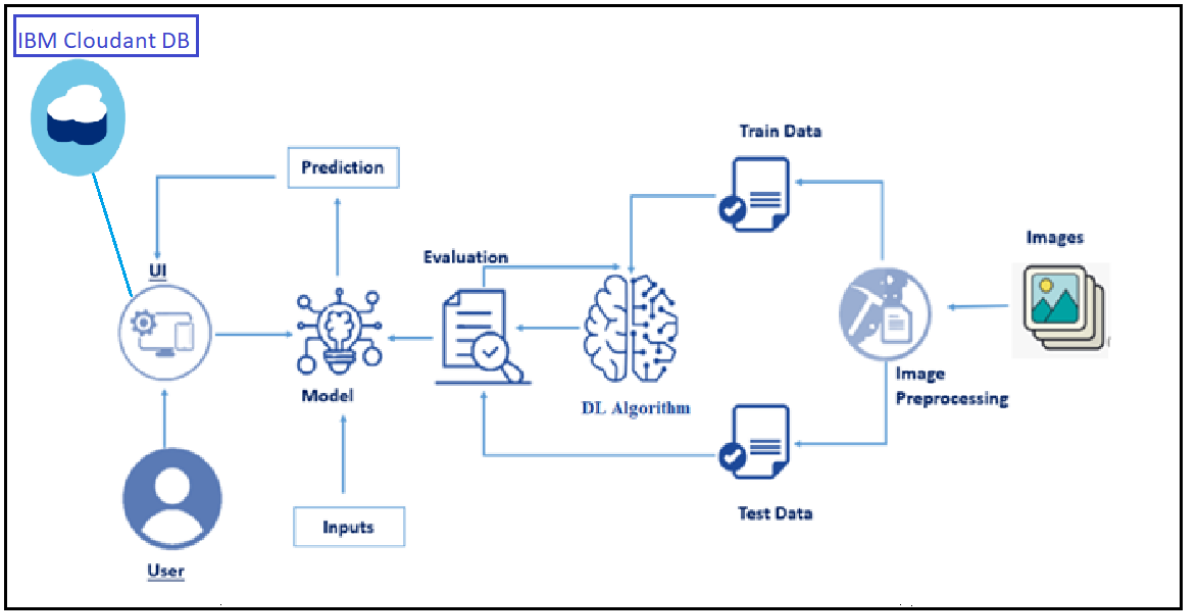
5. PROJECT DESIGN

5.1 Data Flow Diagrams





5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Healthcare Professional)	Screening method	USN-1	As a user, I can find the method more efficient and accurate.	I can reach many patients who could benefit from it.	High	Sprint-1
		USN-2	As a user, I can use it with minimal physical interaction with the device.	It prevents the chances of unwanted infections in the patient's eye.	High	Sprint-2
	Physical features	USN-3	As a user, I can find it portable and light weight.	I can take the device to the residence of patients if they are unable to visit the hospital/clinic.	Low	Sprint-2
	Safety	USN-4	As a user, I can be safe as the detection method is free from radiations.	I can perform the screening procedure without any fear and hesitation.	High	Sprint-4
Customer (Diabetic Patient)	Testing	USN-5	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 the patients from visiting the hospital.	Medium	Sprint-2
		USN-6	As a user, I will be comfortable as it requires minimum/no human involvement.	The screening is carried out using a computer robot along with the aid of AI technology.	Low	Sprint-4
	Results	USN-7	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-8	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-9	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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Public Sector/Private Sector)	Cost-effectiveness	USN-10	As a user, I can reach many people suffering from diabetes.	Diabetic patients are more vulnerable to Diabetic Retinopathy.	Medium	Sprint-1
		USN-11	As a user, I can create awareness among diabetic patients to undergo frequent screening.	As the technique is of low cost, patients will find it very useful.	Low	Sprint-3
	Results	USN-12	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 & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Screening method	USN-1	As a physician, I can find the method more efficient and accurate.	5	High	Dharshine.R.S.
Sprint-2		USN-2	As a patient, I can undergo screening without any physical interaction with my eyes.	5	Medium	Petricia Reshmi.L
Sprint-4		USN-3	As a lab technician, I want the application software to be simple.	5	Low	Dharshine.R.S.
Sprint-4		USN-4	As a patient, I want to get all the updates of testing via the application software.	5	Medium	Petricia Reshmi.L
Sprint-1	Testing	USN-5	As a patient, I can undergo testing without any fear of pain as this method is pain-free.	5	High	Inba Muhil.E.S.
Sprint-3		USN-6	As a patient, I will be comfortable as it requires minimum/no human involvement.	5	Medium	Inba Muhil.E.S
Sprint-1	Results	USN-7	As an ophthalmologist, I can rely on the results without any suspicion.	5	High	Hassim Arsha.S.
Sprint-2		USN-8	As a patient, I can get the results on the spot immediately after the screening process.	5	Low	Inba Muhil.E.S
Sprint-3		USN-9	As a patient, I can benefit from the result as it will help me know whether treatment is necessary or not.	5	High	Dharshine.R.S.
Sprint-4		USN-10	As a patient, I can get the results on the spot immediately after the screening process.	5	Low	Hassim Arsha.S.

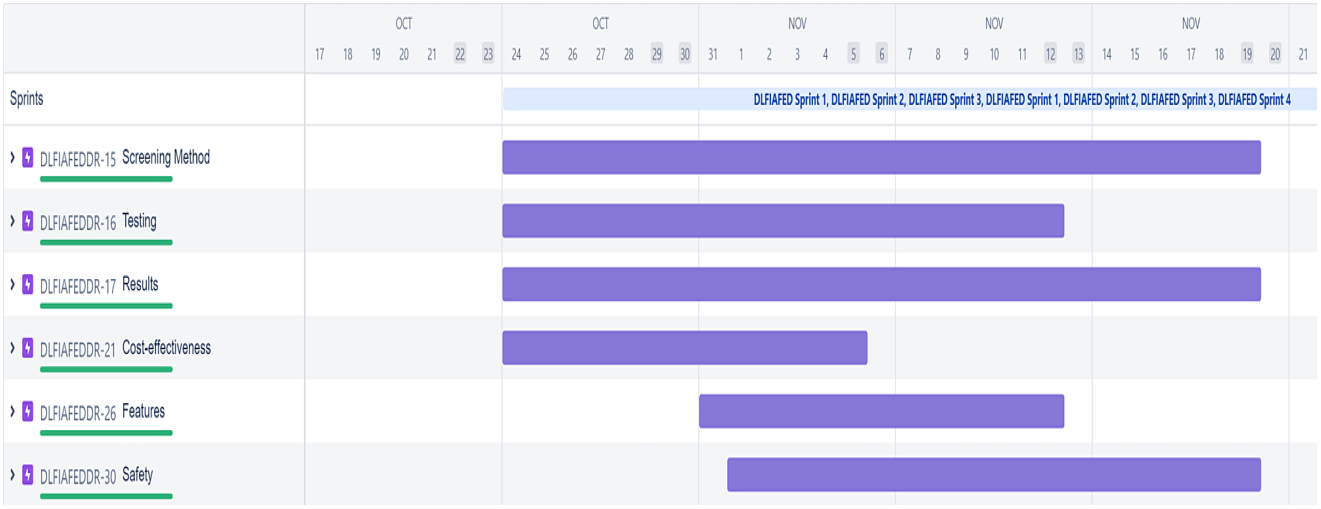
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Features	USN-11	As a lab technician, I want the product to be user-friendly.	5	Medium	Dharshine.R.S.
Sprint-3		USN-12	As a lab technician, I want the steps of the screening process to be easy.	5	High	Petricia Reshmi.L
Sprint-3	Safety	USN-13	As a patient, I can be safe as the detection method is free from radiations.	5	High	Hassim Arsha.S.
Sprint-4		USN-14	As a physician, I want the screening method to follow all the prescribed medical guidelines.	5	Medium	Inba Muhil.E.S.
Sprint-1	Cost-effectiveness	USN-15	As a patient, I want the method to be cost-effective.	5	Medium	Petricia Reshmi.L
Sprint-2		USN-16	As a physician/medical professional, I can create awareness among diabetic patients to undergo frequent screening.	5	Low	Hassim Arsha.S

6.2 Sprint Delivery Schedule

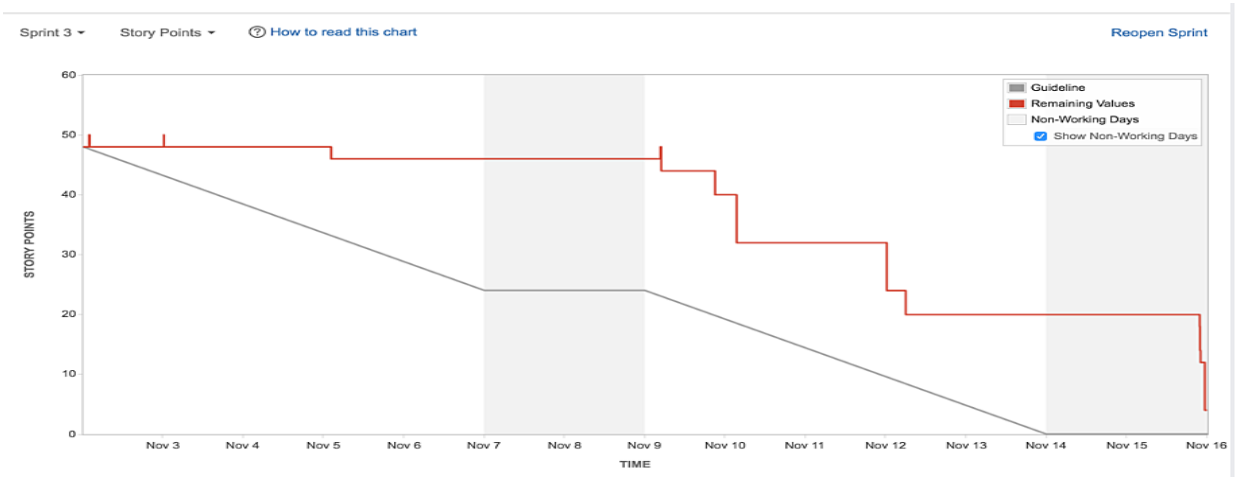
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

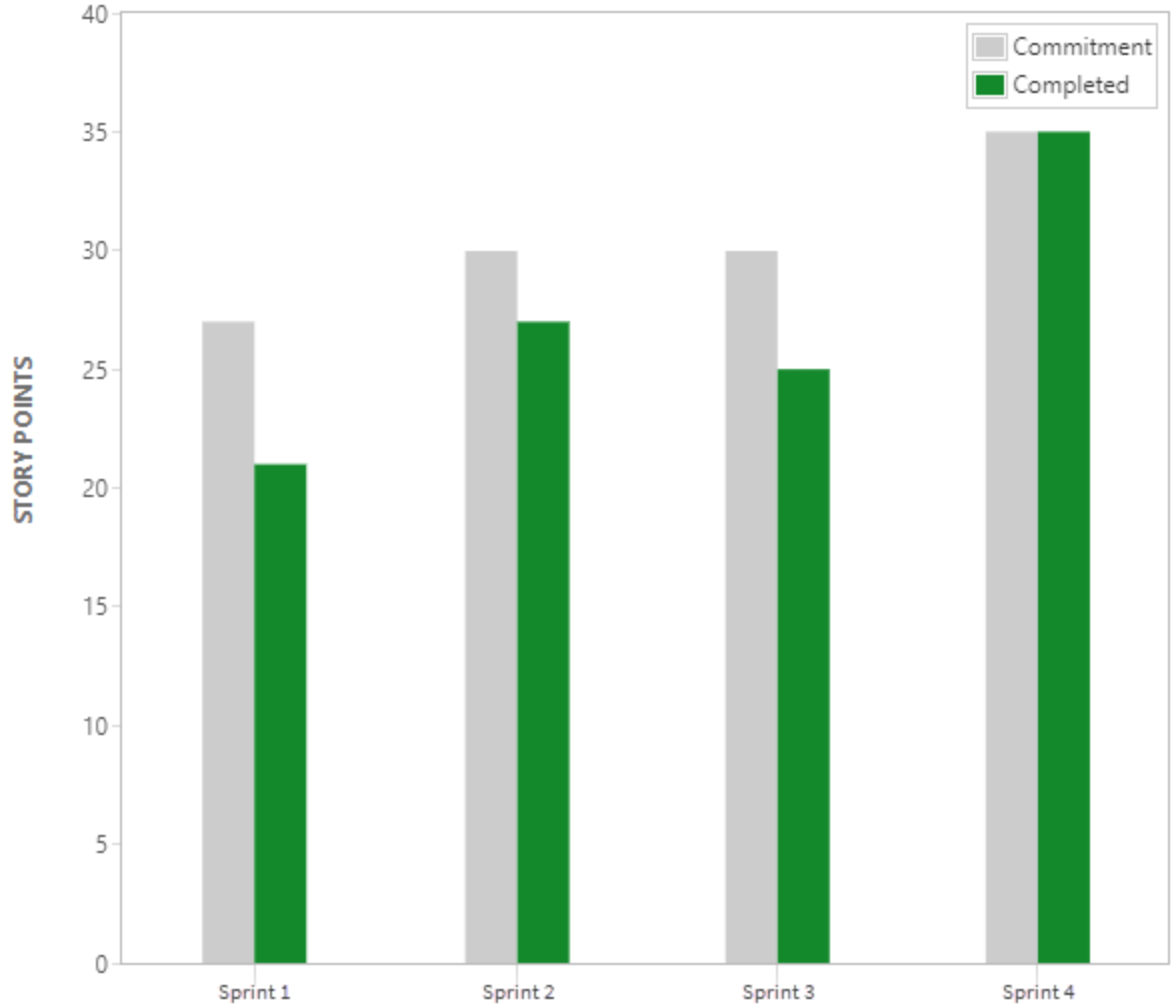
Roadmap



Sprint Burndown chart



Velocity report



7. CODING & SOLUTIONING

7.1 Feature 1

DATA COLLECTION

```
! pip install kaggle

! mkdir ~/.kaggle

! cp kaggle.json ~/.kaggle/

! chmod 600 ~/.kaggle/kaggle.json
```

Downloading the dataset

```
! kaggle datasets download arbethi/diabetic-retinopathy-level-detection
```

Unzipping the dataset

```
! unzip diabetic-retinopathy-level-detection.zip
```

Creating Training And Testing Path

```
imageSize = [299,299]
trainPath = r"/content/preprocessed dataset/preprocessed dataset/training"
testPath = r"/content/preprocessed dataset/preprocessed dataset/testing"
```

DATA PRE-PROCESSING

Importing The 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
```

Configuring 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)
```

Applying ImageDataGenerator Functionality To Train Set And Test Set

```
training_set = train_datagen.flow_from_directory('/content/preprocessed
dataset/preprocessed dataset/training',target_size = (299,299),batch_size =32,
class_mode = 'categorical')
test_set = test_datagen.flow_from_directory('/content/preprocessed
dataset/preprocessed dataset/testing',target_size =(299,299),batch_size =
32,class_mode = 'categorical')
```

MODEL BUILDING

Pre-Training CNN Model As A Feature Extractor

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

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

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

Adding Dense Layers

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

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

model.summary()
Model : "model"
```

Configuring The Learning Process

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

Training The Model

```
# 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
)
```

Saving The Model

```
model.save("diabetic.h5")
```

Output:

```
Found 3662 images belonging to 5 classes.
Found 734 images belonging to 5 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-
applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83683744/83683744 [==============================] - 1s 0us/step
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)

 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)

 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)

 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]']

 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)

 block8_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536
['block8_sepconv3_act[0][0]']
 2D)

 block8_sepconv3_bn (BatchNorma  (None, 19, 19, 728)  2912
['block8_sepconv3[0][0]']
 lization)

 add_6 (Add)                    (None, 19, 19, 728)  0
['block8_sepconv3_bn[0][0]',

'add_5[0][0]']

 block9_sepconv1_act (Activatio  (None, 19, 19, 728)  0
['add_6[0][0]']
 n)

 block9_sepconv1 (SeparableConv  (None, 19, 19, 728)  536536
['block9_sepconv1_act[0][0]']
 2D)

 block9_sepconv1_bn (BatchNorma  (None, 19, 19, 728)  2912
['block9_sepconv1[0][0]']
 lization)

 block9_sepconv2_act (Activatio  (None, 19, 19, 728)  0
['block9_sepconv1_bn[0][0]']
 n)

 block9_sepconv2 (SeparableConv  (None, 19, 19, 728)  536536
```

```
['block9_sepconv2_act[0][0]']
 2D)

 block9_sepconv2_bn (BatchNorma  (None, 19, 19, 728)  2912
['block9_sepconv2[0][0]']
 lization)

 block9_sepconv3_act (Activatio  (None, 19, 19, 728)  0
['block9_sepconv2_bn[0][0]']
 n)

 block9_sepconv3 (SeparableConv  (None, 19, 19, 728)  536536
['block9_sepconv3_act[0][0]']
 2D)

 block9_sepconv3_bn (BatchNorma  (None, 19, 19, 728)  2912
['block9_sepconv3[0][0]']
 lization)

 add_7 (Add)                    (None, 19, 19, 728)  0
['block9_sepconv3_bn[0][0]',

'add_6[0][0]']

 block10_sepconv1_act (Activati  (None, 19, 19, 728)  0
['add_7[0][0]']
 on)

 block10_sepconv1 (SeparableCon  (None, 19, 19, 728)  536536
['block10_sepconv1_act[0][0]']
 v2D)

 block10_sepconv1_bn (BatchNorm  (None, 19, 19, 728)  2912
['block10_sepconv1[0][0]']
 alization)

 block10_sepconv2_act (Activati  (None, 19, 19, 728)  0
['block10_sepconv1_bn[0][0]']
 on)

 block10_sepconv2 (SeparableCon  (None, 19, 19, 728)  536536
['block10_sepconv2_act[0][0]']
 v2D)

 block10_sepconv2_bn (BatchNorm  (None, 19, 19, 728)  2912
['block10_sepconv2[0][0]']
 alization)
```

```
 block10_sepconv3_act (Activati  (None, 19, 19, 728)  0
['block10_sepconv2_bn[0][0]']
 on)

 block10_sepconv3 (SeparableCon  (None, 19, 19, 728)  536536
['block10_sepconv3_act[0][0]']
 v2D)

 block10_sepconv3_bn (BatchNorm  (None, 19, 19, 728)  2912
['block10_sepconv3[0][0]']
 alization)

 add_8 (Add)                    (None, 19, 19, 728)  0
['block10_sepconv3_bn[0][0]',

'add_7[0][0]']

 block11_sepconv1_act (Activati  (None, 19, 19, 728)  0
['add_8[0][0]']
 on)

 block11_sepconv1 (SeparableCon  (None, 19, 19, 728)  536536
['block11_sepconv1_act[0][0]']
 v2D)

 block11_sepconv1_bn (BatchNorm  (None, 19, 19, 728)  2912
['block11_sepconv1[0][0]']
 alization)

 block11_sepconv2_act (Activati  (None, 19, 19, 728)  0
['block11_sepconv1_bn[0][0]']
 on)

 block11_sepconv2 (SeparableCon  (None, 19, 19, 728)  536536
['block11_sepconv2_act[0][0]']
 v2D)

 block11_sepconv2_bn (BatchNorm  (None, 19, 19, 728)  2912
['block11_sepconv2[0][0]']
 alization)

 block11_sepconv3_act (Activati  (None, 19, 19, 728)  0
['block11_sepconv2_bn[0][0]']
 on)

 block11_sepconv3 (SeparableCon  (None, 19, 19, 728)  536536
['block11_sepconv3_act[0][0]']
 v2D)
```

```
 block11_sepconv3_bn (BatchNorm  (None, 19, 19, 728)  2912
['block11_sepconv3[0][0]']
 alization)

 add_9 (Add)                    (None, 19, 19, 728)  0
['block11_sepconv3_bn[0][0]',

'add_8[0][0]']

 block12_sepconv1_act (Activati  (None, 19, 19, 728)  0
['add_9[0][0]']
 on)

 block12_sepconv1 (SeparableCon  (None, 19, 19, 728)  536536
['block12_sepconv1_act[0][0]']
 v2D)

 block12_sepconv1_bn (BatchNorm  (None, 19, 19, 728)  2912
['block12_sepconv1[0][0]']
 alization)

 block12_sepconv2_act (Activati  (None, 19, 19, 728)  0
['block12_sepconv1_bn[0][0]']
 on)

 block12_sepconv2 (SeparableCon  (None, 19, 19, 728)  536536
['block12_sepconv2_act[0][0]']
 v2D)

 block12_sepconv2_bn (BatchNorm  (None, 19, 19, 728)  2912
['block12_sepconv2[0][0]']
 alization)

 block12_sepconv3_act (Activati  (None, 19, 19, 728)  0
['block12_sepconv2_bn[0][0]']
 on)

 block12_sepconv3 (SeparableCon  (None, 19, 19, 728)  536536
['block12_sepconv3_act[0][0]']
 v2D)

 block12_sepconv3_bn (BatchNorm  (None, 19, 19, 728)  2912
['block12_sepconv3[0][0]']
 alization)

 add_10 (Add)                   (None, 19, 19, 728)  0
['block12_sepconv3_bn[0][0]',
```

```
'add_9[0][0]']

 block13_sepconv1_act (Activati  (None, 19, 19, 728)  0
['add_10[0][0]']
 on)

 block13_sepconv1 (SeparableCon  (None, 19, 19, 728)  536536
['block13_sepconv1_act[0][0]']
 v2D)

 block13_sepconv1_bn (BatchNorm  (None, 19, 19, 728)  2912
['block13_sepconv1[0][0]']
 alization)

 block13_sepconv2_act (Activati  (None, 19, 19, 728)  0
['block13_sepconv1_bn[0][0]']
 on)

 block13_sepconv2 (SeparableCon  (None, 19, 19, 1024  752024
['block13_sepconv2_act[0][0]']
 v2D)                           )

 block13_sepconv2_bn (BatchNorm  (None, 19, 19, 1024  4096
['block13_sepconv2[0][0]']
 alization)                     )

 conv2d_3 (Conv2D)              (None, 10, 10, 1024  745472
['add_10[0][0]']
                                )

 block13_pool (MaxPooling2D)    (None, 10, 10, 1024  0
['block13_sepconv2_bn[0][0]']
                                )

 batch_normalization_3 (BatchNo  (None, 10, 10, 1024  4096
['conv2d_3[0][0]']
 rmalization)                   )

 add_11 (Add)                   (None, 10, 10, 1024  0
['block13_pool[0][0]',           ['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
__________________________________________________________________________________
____________________
Epoch 1/30
3/3 [==============================] - 55s 17s/step - loss: 11.8808 -
accuracy: 0.3854
Epoch 2/30
3/3 [==============================] - 43s 12s/step - loss: 12.2067 -
accuracy: 0.5625
Epoch 3/30
3/3 [==============================] - 47s 13s/step - loss: 9.0027 - accuracy:
0.5833
Epoch 4/30
3/3 [==============================] - 46s 13s/step - loss: 6.3194 - accuracy:
0.5729
Epoch 5/30
3/3 [==============================] - 45s 13s/step - loss: 5.9865 - accuracy:
0.5625
Epoch 6/30
3/3 [==============================] - 44s 13s/step - loss: 4.8598 - accuracy:
0.7083
Epoch 7/30
3/3 [==============================] - 44s 13s/step - loss: 4.9243 - accuracy:
0.5938
Epoch 8/30
3/3 [==============================] - 43s 13s/step - loss: 5.4845 - accuracy:
0.6250
```

```
Epoch 9/30
3/3 [==============================] - 43s 13s/step - loss: 4.2899 - accuracy:
0.6146
Epoch 10/30
3/3 [==============================] - 44s 13s/step - loss: 3.8048 - accuracy:
0.6458
Epoch 11/30
3/3 [==============================] - 44s 13s/step - loss: 3.5794 - accuracy:
0.6250
Epoch 12/30
3/3 [==============================] - 45s 13s/step - loss: 3.4683 - accuracy:
0.6458
Epoch 13/30
3/3 [==============================] - 48s 15s/step - loss: 2.8153 - accuracy:
0.7917
Epoch 14/30
3/3 [==============================] - 45s 13s/step - loss: 3.0900 - accuracy:
0.6458
Epoch 15/30
3/3 [==============================] - 47s 13s/step - loss: 2.9761 - accuracy:
0.6875
Epoch 16/30
3/3 [==============================] - 44s 13s/step - loss: 2.6205 - accuracy:
0.7292
Epoch 17/30
3/3 [==============================] - 44s 13s/step - loss: 2.8528 - accuracy:
0.6458
Epoch 18/30
3/3 [==============================] - 44s 13s/step - loss: 2.7091 - accuracy:
0.6458
Epoch 19/30
3/3 [==============================] - 45s 13s/step - loss: 3.1372 - accuracy:
0.6667
Epoch 20/30
3/3 [==============================] - 45s 13s/step - loss: 2.6082 - accuracy:
0.7500
Epoch 21/30
3/3 [==============================] - 44s 13s/step - loss: 3.9316 - accuracy:
0.7188
Epoch 22/30
3/3 [==============================] - 44s 13s/step - loss: 3.0207 - accuracy:
0.7292
Epoch 23/30
3/3 [==============================] - 48s 14s/step - loss: 2.7646 - accuracy:
0.6250
Epoch 24/30
3/3 [==============================] - 36s 13s/step - loss: 3.3575 - accuracy:
0.7308
```

```
Epoch 25/30
3/3 [==============================] - 53s 14s/step - loss: 3.2141 - accuracy:
0.7396
Epoch 26/30
3/3 [==============================] - 48s 15s/step - loss: 3.9002 - accuracy:
0.6979
Epoch 27/30
3/3 [==============================] - 45s 13s/step - loss: 2.6503 - accuracy:
0.7083
Epoch 28/30
3/3 [==============================] - 44s 13s/step - loss: 2.6253 - accuracy:
0.7812
Epoch 29/30
3/3 [==============================] - 45s 13s/step - loss: 3.3202 - accuracy:
0.6771
Epoch 30/30
3/3 [==============================] - 44s 13s/step - loss: 4.4876 - accuracy:
0.6667
```

7.2 Feature 2

Application building (Using Python-Flask)

```
import numpy as np
import os
from tensorflow import keras

from keras import models
from keras.models import load_model
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input
import requests
from flask import Flask, request, render_template, redirect, url_for
from cloudant.client import Cloudant

model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
app = Flask(_name_)
# Authenticate using an IAM API key
client = Cloudant.iam('367e91e7-6150-4f63-92f4-24625af53457-bluemix',
                      'EqKm5BOKxzGLIm9YsFnXKJ66ywOyL9tDVK9oN0_FPD4G', connect=True)
# Create a database using an initialized client
my_database = client.create_database('my_db')
if my_database.exists():
   print("Database '{0}' successfully created.".format('my_db'))
```

```
# default home page or route
@app.route('/')
def index():
    return render_template('index.html')


@app.route('/index')
def home():
    return render_template("index.html")


'"@ app.route('/register')
def register():
    return render_template("register.html")""


# registration page
@app.route('/register', methods=["GET", "POST"])
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 == "POST":
        user = request.form.get('name')
        passw = request.form.get('pass')
        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]['pswd'])):
                return redirect(url_for('prediction'))
            else:
                print('Invalid User')
    else:
        return render_template('login.html')


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


@app.route("/predict")
def predict():
    return render_template("prediction.html")


@app.route('/result', methods=["GET", "POST"])
def res():
    if request.method == "POST":
        f = request.files['image']
        # 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(basepath, 'uploads', 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)
    # 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 [ 011 )
    result = str(index[prediction[0]])
    print(result)
    return render_template('prediction.html', prediction=result)


if _name_ == "_main_":
   app.run(debug=False)
```

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button	User is authorized and has an account.	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://drdetection.com	Login/Signup popup should display	Working as expected	Pass	Steps are clear to follow	Y	-	Dharshine.R.S.
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup	Verify user is able to search by entering keywords in search box	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://drdetection.com	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	Steps are clear to follow	Y	-	Hassim Aisha.S.
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials	Verify user is able to see suggestions based on keyword entered in search box	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	Pass	Steps are clear to follow	Y	-	Inba Muhil.S.
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials	Verify user is able to see related auto suggestions displaying based on keyword entered in search box	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Steps are clear to follow	Y	-	Petricia Reshmi.L.
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials	Verify user is able to see no matches found message when no results are matching with entered keyword	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam@gmail.com password: Testing12367866876876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Steps are clear to follow	Y	-	Dharshine.R.S.
LoginPage_TC_005	Functional	Login page	Verify user is able to log into application with InValid credentials	Verify user is able to see seach detailed page when nothing entered in textbox.	1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in	Username: chalam password: Testing12367866876876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	Steps are clear to follow	Y	-	Petricia Reshmi.L.

8.2 User Acceptance Testing

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

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	5	0	0	5
Client Application	10	0	0	10
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

Model Summary

```
[16] model.summary()
     Model : "model"
```

```
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 (BatchNormalization)  (None, 149, 149, 32)  128

 block1_conv1_act (Activation)  (None, 149, 149, 32)  0

 block1_conv2 (Conv2D)       (None, 147, 147, 64)      18432

 block1_conv2_bn (BatchNormalization)  (None, 147, 147, 64)  256

 block1_conv2_act (Activation)  (None, 147, 147, 64)  0

 block2_sepconv1 (SeparableConv2D)  (None, 147, 147, 128)  8768

 block2_sepconv1_bn (BatchNormalization)  (None, 147, 147, 128)  512

 block2_sepconv2_act (Activation)  (None, 147, 147, 128)  0
```

.

.

.

```
 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
__________________________________________________________________________________________________
```

Accuracy (Training and 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
```

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

1) Early detection and treatment can reduce the risk of vision loss for diabetics by 25%.
2)The economic cost of early diagnosis and treatment is low.
3)Treatment for patients with recognizable disease is safe, effective and universally agreeable.
4) It is an appropriate screening procedure which is acceptable both to the public and health care professionals.

10.2 Disadvantages

1) It requires very large amount of data in order to perform better than other techniques.

2)It is extremely expensive to train due to complex data models. Moreover deep learning requires expensive GPUs and hundreds of machines. This increases cost to the users.

3)There is no standard theory to guide you in selecting right deep learning tools as it requires knowledge of topology, training method and other parameters. As a result it is difficult to be adopted by less skilled people.

11. CONCLUSION

Diabetic retinopathy is a serious complication of diabetes mellitus, leading to progressive damage and even blindness of the retina. Its early detection and treatment is important in order to prevent its deterioration and the retina's damage. The interest in applying deep learning in detecting diabetic retinopathy has increased during the past years and as several DL systems evolve and become integrated into the clinical practice, they will enable the clinicians to treat the patients in need more effectively and efficiently. This article presents the current state of research regarding the application of deep learning in diagnosing diabetic retinopathy. Although deep learning has paved the way for more accurate diagnosis and treatment, further improvements are still necessary regarding performance, interpretability and trustworthiness from ophthalmologists.

12. FUTURE SCOPE

Use of AI in medical diagnostics, especially in ophthalmology heralds a new era. If proven to be sensitive and specific enough this technology can totally change the way we look at screening programs and community-based ophthalmology programs. Most of the present systems use conventional of 30–50° fundus images.A lot of work is also being done on identifying serum biomarkers for early detection and monitoring of diseases like diabetic retinopathy. Thus, a comprehensive analysis of fundus imaging, systemic parameter profile and other serum biomarkers using AI might provide better insights, perhaps even better conclusions than what human intelligence is capable of deriving.

13. APPENDIX

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-40335-1660628291

Team ID: PNT2022TMID34131