A Project report submitted in-partial fulfillment of the 7th semester in degree of

BACHELOR OF ENGINEERING
IN

Electronics and Communication Engineering

Submitted by

TeamID: PNT2022TMID24435

BALASUBRAMANI M	113319106007
SANTHANU K C	113319106067
DHAKSHANAMOORTHY M	113319106017
JOSH ADITHEA R S	113319106033

Department Of Electronics and Communication Engineering

ANNA UNIVERSITY : CHENNAI - 600025

NOV-2022

Velammal Institute of Technology

(A Affiliated College of Anna University ,Chennai)

BONAFIDECERTIFICATE

CERTIFIED THAT THIS PROJECT REPORT "Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy**" IS THE BONAFIDE RECORD WORK DONE BY MR .M.BALASUBRAMANI(113319106007),MR.K.C.SANTHANU(113319106067), MR.M.DHAKSHANAMOORTHY (113319106017) AND MR.R.S.JOSH ADITHEA(113319106033) FOR "HX8001 PROFESSIONAL READINESS FOR INNOVATION , EMPLOYABILITY AND ENTREPRENEURSHIP " IN VII SEMESTER OF B.E., DEGREE COURSE IN ELECTRONICS AND COMMUNICATION ENGINEERING BRANCH DURING THE ACADEMIC YEAR OF 2022-2023.**

Staff-Incharge	***Evaluator***
Mrs.Sivarathna	**Dr.Sridevi**

AKNOWLEDGEMENT

The satisfactions that accompany the successful completion of any task would be incomplete without mentioning the people who made it possible by their constant guidance and encouragement crowned our efforts with success. We are grateful to our Chairman Shri. M.V.Muthuramalingam and our Director Shri. M.V.M.Sasikumar for facilitating us with this opportunity. Our sincere thanks to Advisor's Shri. K.Razak and Shri.M. Vasu ,Principal Dr.N.Balaji ,Vice Principal Dr. S. Soundarajan for their support. Our respected Head of the Department of Electronics and Communication Engineering Dr.B. Sridevi, Project Evaluator Dr. B.Sridevi and Dr.Karthikeyan and our Industrial mentor Sai Priya deserve a special note of thanks and gratitude for having extended their fullest cooperation and continuous suggestions to make this project successful. We also thank all the faculty members of Electronics and Communication Engineering department for their help during the course of this project. We also like to express our gratefulness to our beloved parents and our family members who have always provided backup with their unending moral support and of course momentary help too which we believe are the driving forces for the completion of the project.

Project Report Format

1. **INTRODUCTION**
   - 1.1 Project Overview
   - 1.2 Purpose
2. **LITERATURE SURVEY**
   - 2.1 Existing problem
   - 2.2 References
   - 2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   - 3.1 Empathy Map Canvas
   - 3.2 Ideation & Brainstorming
   - 3.3 Proposed Solution
   - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   - 4.1 Functional requirement
   - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   - 5.1 Data Flow Diagrams
   - 5.2 Solution & Technical Architecture
   - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   - 6.1 Sprint Planning & Estimation
   - 6.2 Sprint Delivery Schedule
   - 6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   - 7.1 Feature 1
   - 7.2 Feature 2
   - 7.3 Database Schema (if Applicable)
8. **TESTING**
   - 8.1 Test Cases
   - 8.2 User Acceptance Testing
9. **RESULTS**
   - 9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

    Source Code

    GitHub & Project Demo Link

INTRODUCTION:

PROJECT OVERVIEW:

Diabetic retinopathy (die-uh-BET-ik ret-ih-NOP-uh-thee) 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.

PURPOSE:

Diabetic retinopathy is caused by **high blood sugar due to diabetes**. Over time, having too much sugar in your blood can damage your retina — the part of your eye that detects light and sends signals to your brain through a nerve in the back of your eye (optic nerve). Diabetes damages blood vessels all over the body.

In the early stages of diabetic retinopathy, your eye doctor will probably just keep track of how your eyes are doing. Some people with diabetic retinopathy may need a comprehensive dilated eye exam as often as every 2 to 4 months.

In later stages, it’s important to start treatment right away — especially if you have changes in your vision. While it won’t undo any damage to your vision, treatment can stop your vision from getting worse. It’s also important to take steps to control your diabetes, blood pressure, and cholesterol.

LITERATURE SURVEY:

EXISTING PROBLEM:

Customer Problem Statement Template:
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.

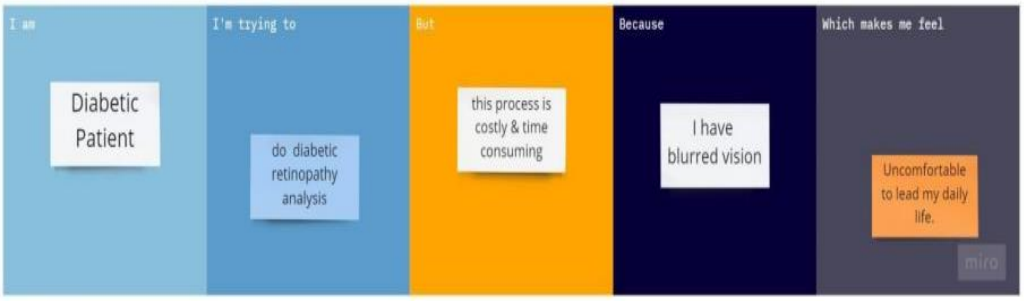
REFERENCES:

A. Bali and V. Mansotra, "Deep Learning-based Techniques for the Automatic Classification of Fundus Images: A Comparative Study," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021, pp. 351-359, doi: 10.1109/ICAC3N53548.2021.9725464.
W. Zhang, X. Zhao, Y. Chen, J. Zhong and Z. Yi, "DeepUWF: An Automated Ultra-Wide-Field Fundus Screening System via Deep Learning," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 8, pp. 2988-2996, Aug. 2021, doi: 10.1109/JBHI.2020.3046771.
H. Kaushik, D. Singh, M. Kaur, H. Alshazly, A. Zaguia and H. Hamam, "Diabetic Retinopathy Diagnosis From Fundus Images Using Stacked Generalization of Deep Models," in IEEE Access, vol. 9, pp. 108276-108292, 2021, doi: 10.1109/ACCESS.2021.3101142.
H. Yeh, C. -J. Lin, C. -C. Hsu and C. -Y. Lee, "Deep-learning based automated segmentation of Diabetic Retinopathy symptoms," 2020 International Symposium on Computer, Consumer and Control (IS3C), 2020, pp. 497-499, doi: 10.1109/IS3C50286.2020.00135.
B. Goutam, M. F. Hashmi, Z. W. Geem and N. D. Bokde, "A Comprehensive Review of Deep Learning Strategies in Retinal Disease Diagnosis Using Fundus Images," in IEEE Access, vol. 10, pp. 57796-57823, 2022, doi: 10.1109/ACCESS.2022.3178372.
N. Memari, S. Abdollahi, M. M. Ganzagh and M. Moghbel, "Computer-assisted diagnosis (CAD) system for Diabetic Retinopathy screening using color fundus images using Deep learning," 2020 IEEE Student Conference on Research and Development (SCOReD), 2020, pp. 69-73, doi: 10.1109/SCOReD50371.2020.9250986.
B. Bulut, V. Kalın, B. B. Güneş and R. Khazhin, "Deep Learning Approach For Detection Of Retinal Abnormalities Based On Color Fundus Images," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259870.
M. Z. Atwany, A. H. Sahyoun and M. Yaqub, "Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey," in IEEE Access, vol. 10, pp. 28642-28655, 2022, doi: 10.1109/ACCESS.2022.3157632.

PROBLEM STATEMENT DEFNITOIN:

Customer Problem Statement Template:
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.

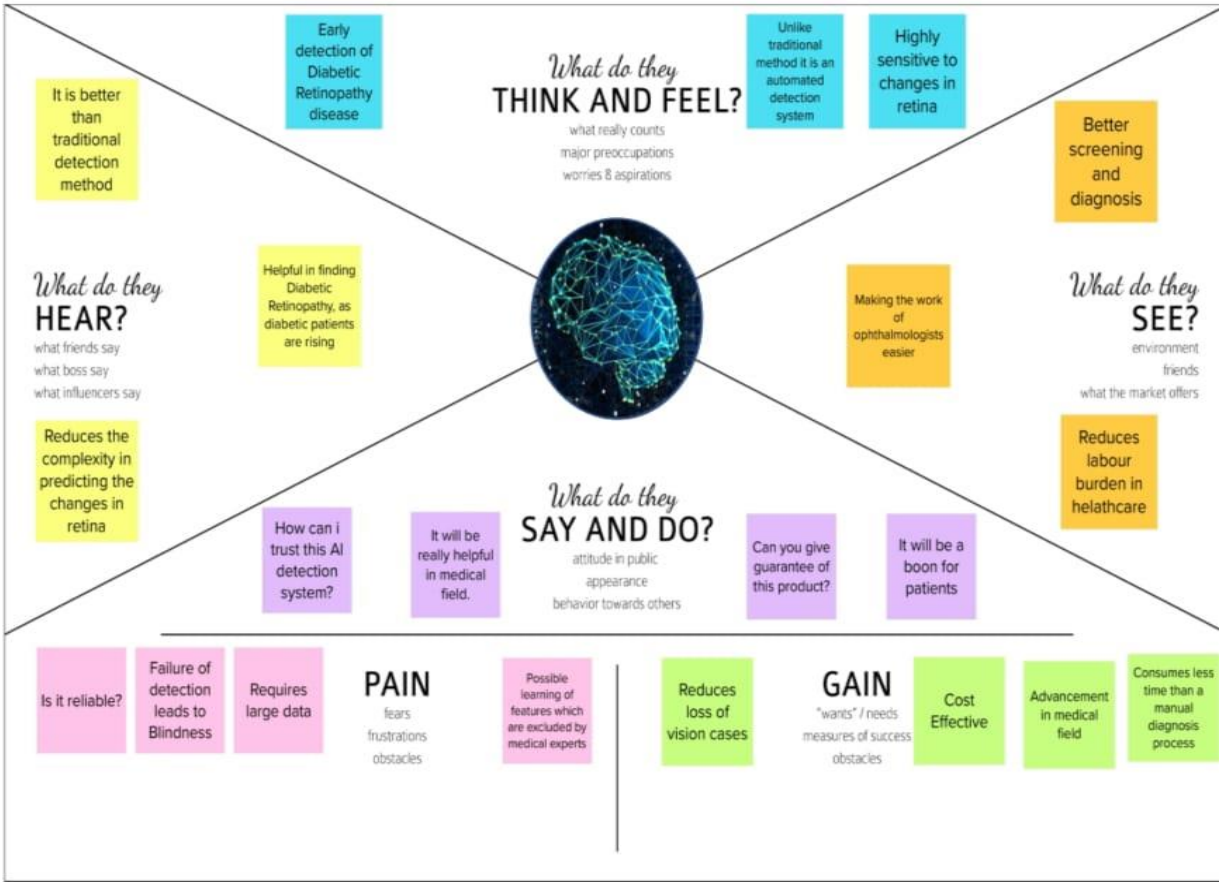


Patient 2:-

IDEATION AND PROPOSDED SOLUTION:

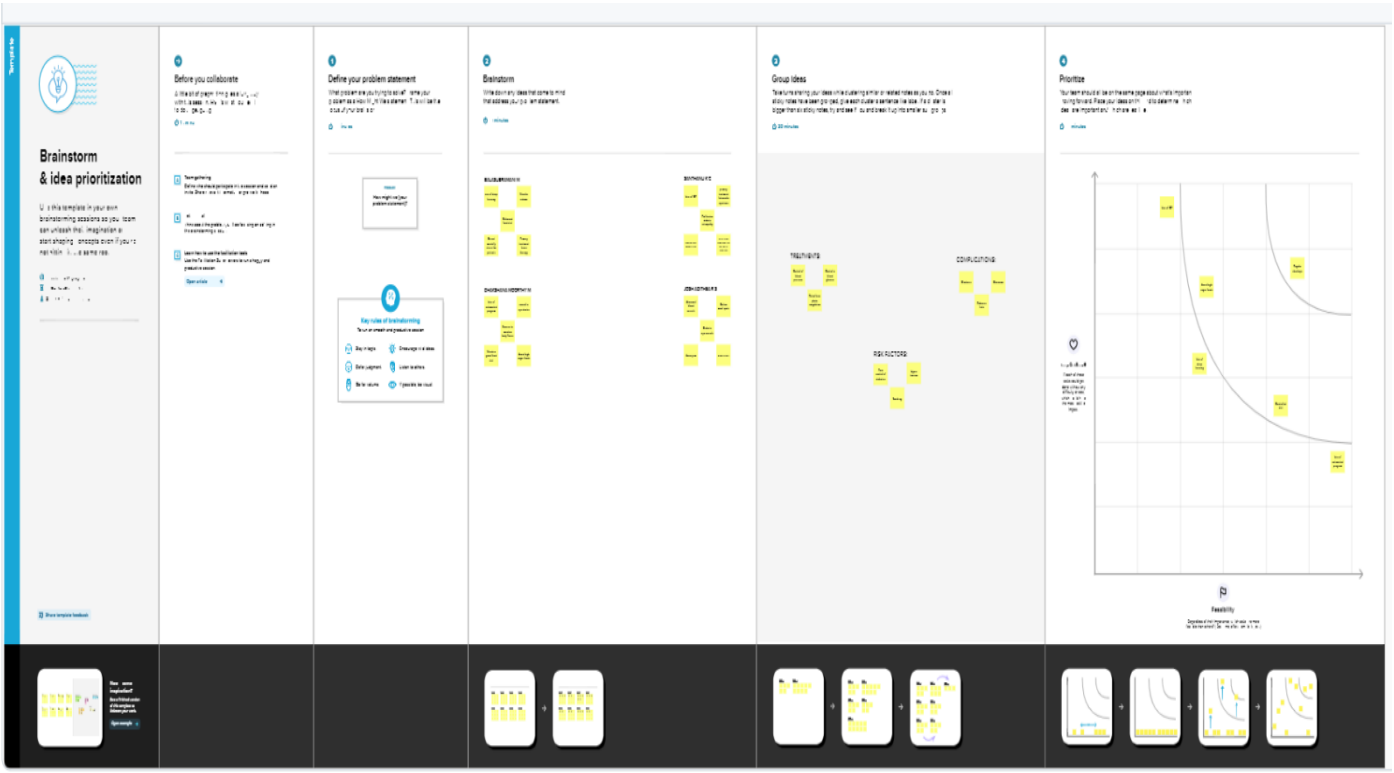
EMPATHY MAP CANVAS:

Diabetes is a serious complex condition affecting millions of people every year. It is a condition that requires constant monitoring and if complications develop it can reduce life expectancy. Moreover there is no cure for diabetes. Diabetic Retinopathy is a common complication of diabetes, which causes lesions on retina that affect vision. If it is not detected early it can lead to blindness. Our project is to detect Diabetic Retinopathy Using Deep Learning.



IDEATION AND BRAINSTORMONG:

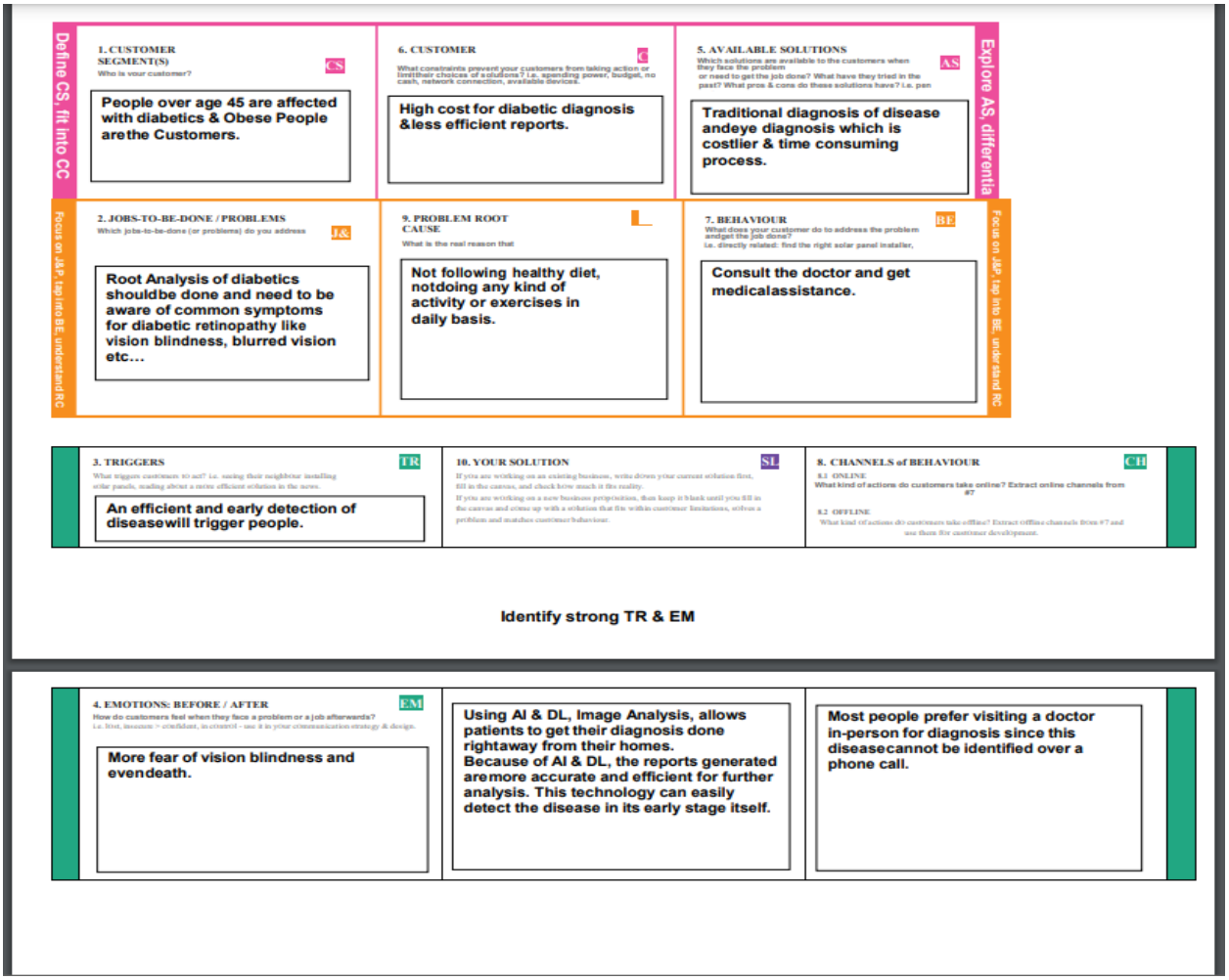
Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina). At first, diabetic retinopathy might cause no symptoms or only mild vision problems. But it can lead to blindness.The condition 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.



PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Low cost Eye testing and efficient results, like computer aided testing.
2.	Idea / Solution description	AI has been applied to image based medical field and suitable for processing complex image AI can assist diabetic retinopathy
3.	Novelty / Uniqueness	Owing the characteristics of DL the training set have abundant and high quality images. Which will be easy and appropriate to evaluate
4.	Social Impact / Customer Satisfaction	Quality of life of the diabetics with retinopathy Can be improved and they can lead a life with colour full vision
5.	Business Model (Revenue Model)	As the testing and analysing is taken under low cost this can be developed into a business model so that it can help both the patient and the developer
6.	Scalability of the Solution	Cost of the diabetic retinopathy treatment is about 60,000-1,00,000 which can be reduced effectively which can be helpful for the patient and the product developer can also gain profit as the invested cost is also low

PROBLEM SOLUTION FIT:



Identify strong TR & EM

REQUIREMENTS ANALYSIS:

FUNCTIONAL REQUIREMENT:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identify and select dataset	The appropriate dataset to enhance the model's performance is 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 truth and false of the medical condition [Diabetic Retinopathy].
FR-4	Analysis	Based on the training the model should analyze the medical condition [DR] to predict/detect thedisease accurately.
FR-5	Testing	The trained model is tested with different data to ensure it has trained well to predict/detect the medical condition [DR].
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:

Non-functional Requirements:

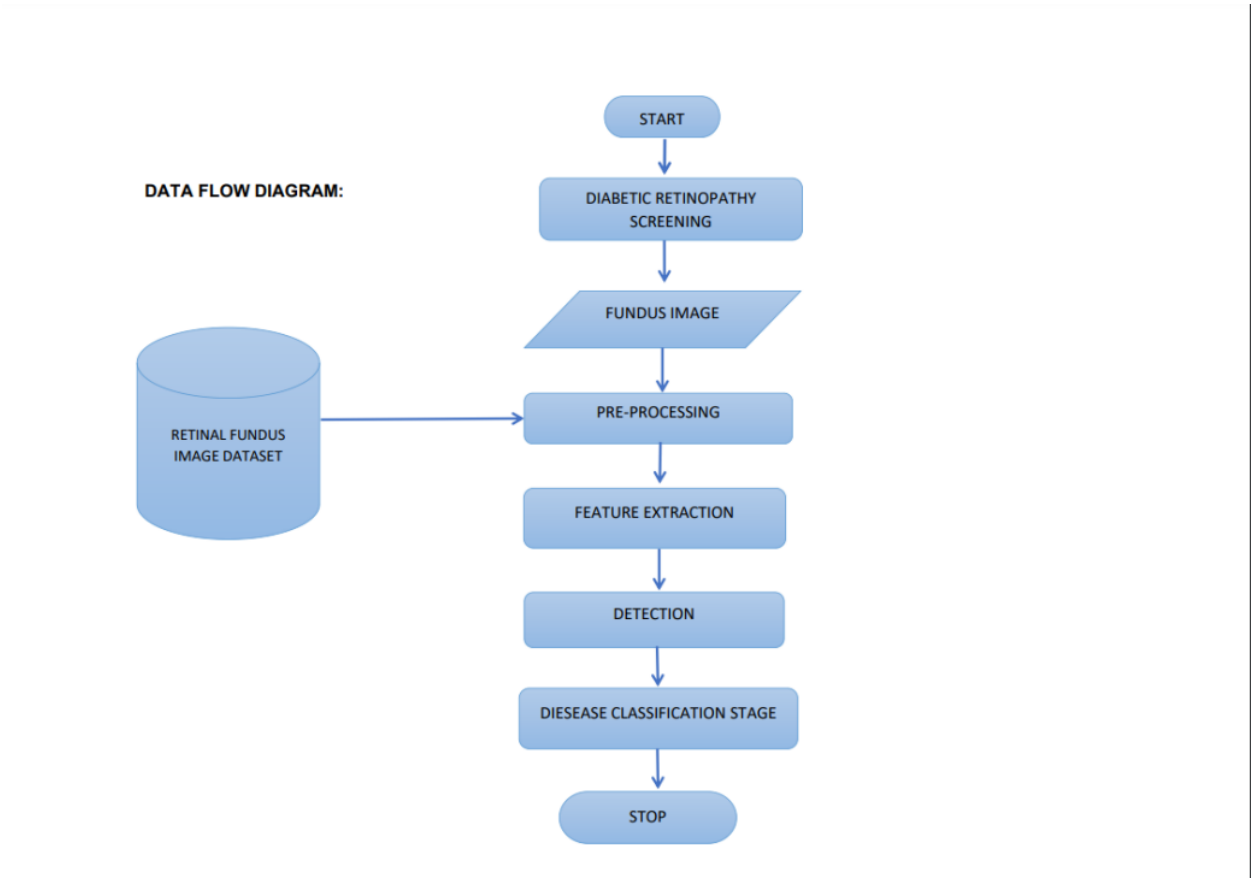
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User with a basic understanding of the medical condition and computer knowledge can operate the system. User-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 of different then 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 a low cost sothat everyone with DR can find it beneficial.
NFR-5	**Scalability**	By processing more datasets for the reference of DR detection

PROJECT DESIGN:

DATA FLOW DIAGRAM:

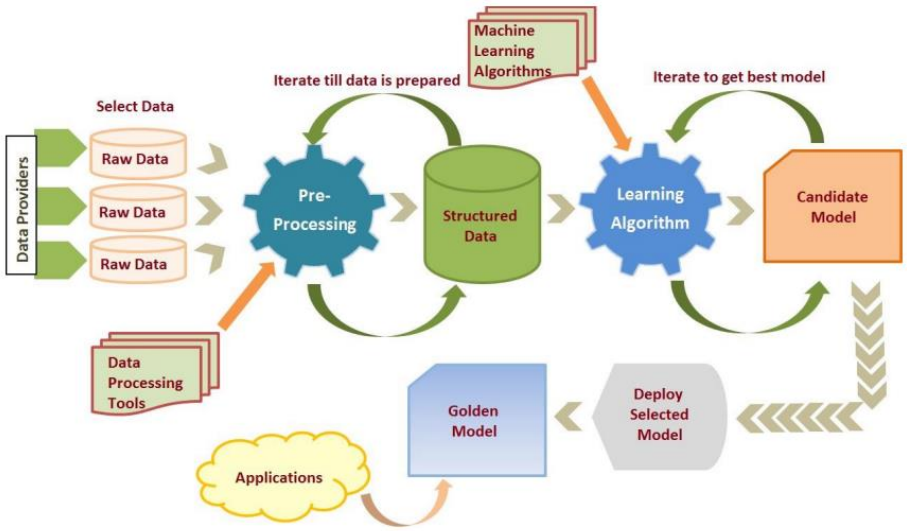
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



SOLUTION AND TECHNICAL ARCHITECTURE:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to: • Find the best tech solution to solve existing business problems. • Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders. • Define features, development phases, and solution requirements. • Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:



USER STORIES:

User Stories

Use the below template to list all the user stories for the product.

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 or not by uploading the image of my eye by entering details.	I can upload or take image.	High	Sprint-1
	Screening method	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 use it with minimal physical interaction with the device.	I can take the device to the residence of patients if they are unable to visit the hospital/clinic.	High	Sprint-2
	Physical feature	USN-4	As a user, I can find it portable and light weight.	I can perform the screening procedure without any fear and hesitation.	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 the 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 the patients from visiting the hospital.	Medium	Sprint-2
		USN-7	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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	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 Diabetic Retinopathy.	Medium	Sprint-1
		USN-12	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-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

PROJECT PLANNING AND SCHEDULING:

SPRINT PLANNING AND ESTIMATION:

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

Use the below template to create a product backlog and sprint schedule

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, and password, and confirming my password.	10	High	Balasubramani, Joshadithea
Sprint-1	E-Mail Conformation	USN-2	As a user, i will receive a confirmation email oncel have registered for the application	10	Medium	Balasubramani, Santhanu
Sprint-2	Login	USN-3	As a user, I can log into the application by entering my email & password	5	Low	Balasubramani, Joshadithea
Sprint-2	Upload Images	USN-4	As a user, I Will upload the images for further analysis	10	High	Dhakshanamoorthy, Santhanu
Sprint-2	Dashboard	USN-5	As a user, I can navigate through the dashboard	5	Low	Balasubramani, Joshadithea
Sprint-3	Train the Model	Task – 1	Creating a dataset based on images & Creating & training the model created using the dataset	20	High	Dhakshanamoorthy, Santhanu
Sprint-4	Testing & Evaluation	Task – 2	As a developer. Testing the model with predicted results & evaluating the results	10	High	Dhakshanamoorthy, Joshadithea
Sprint-4	Display predicted results	USN - 6	Displaying the results on the dashboard and measures to be taken.	10	High	Dhakshanamoorthy, Santhanu

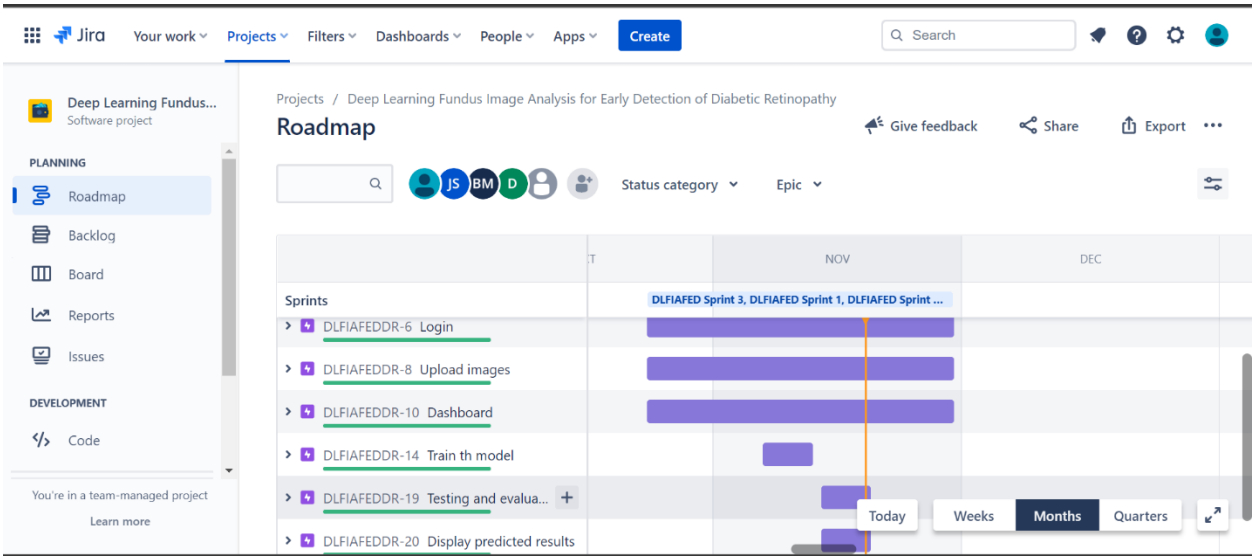
SPRINT DELIVERY SCHEDULE:

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

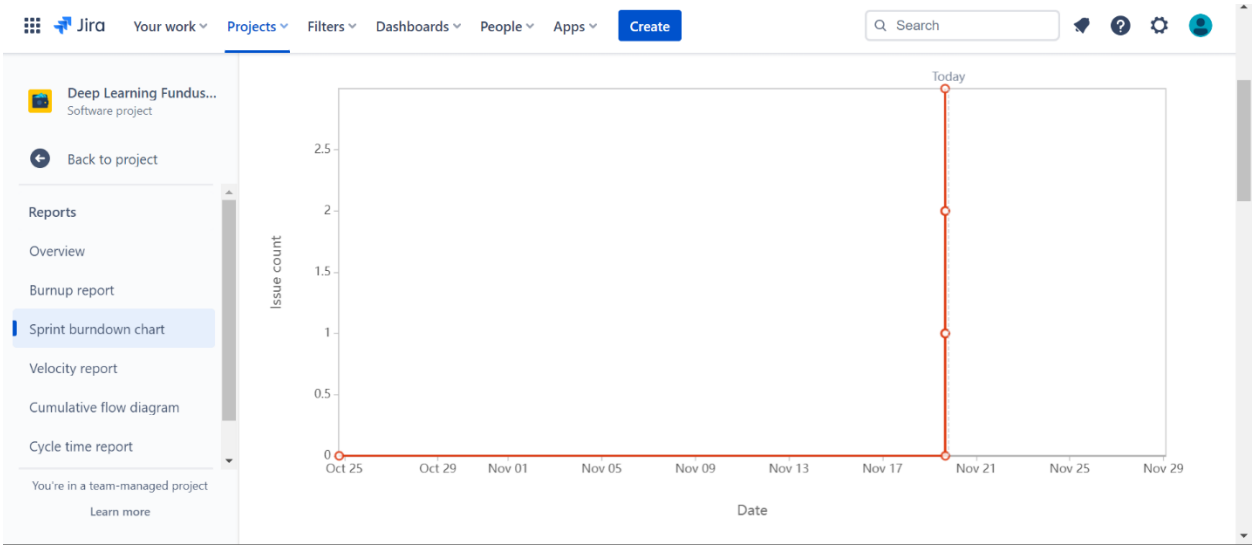
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

REPORTS FROM JIRA:

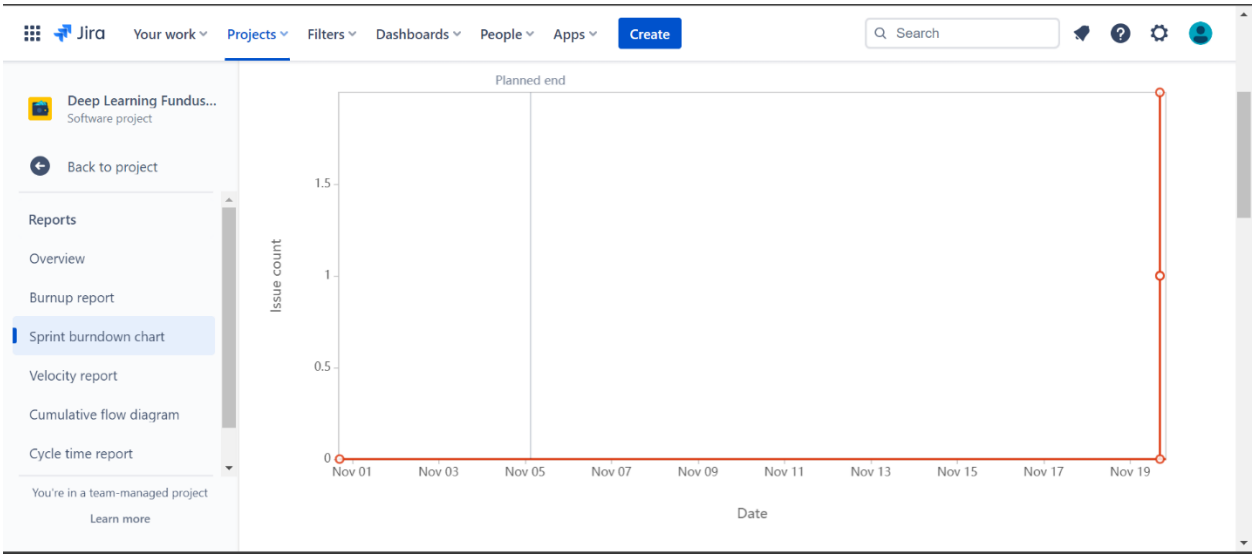
ROADMAP:



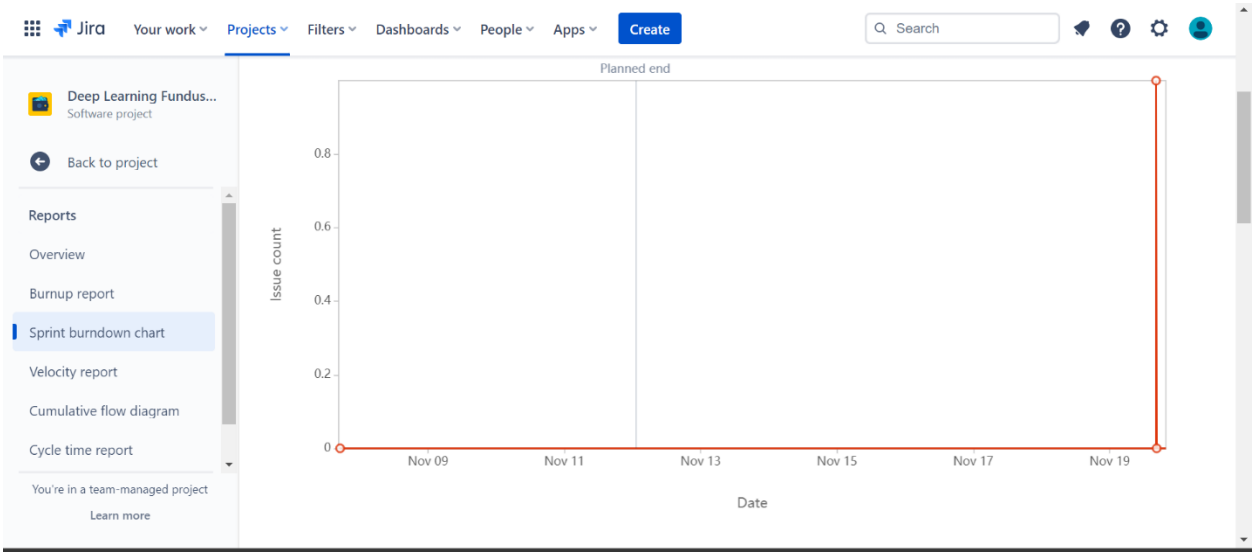
SPRINT1:



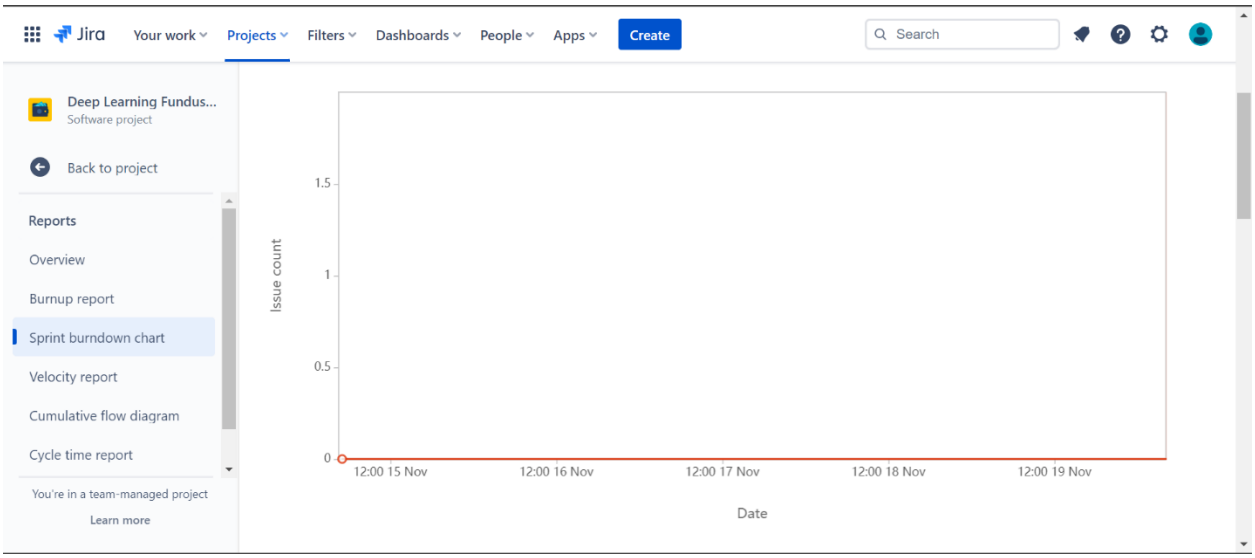
SPRINT2:



SPRINT3:



SPRINT4:



CODING AND SOLUTIONING:

Feature 1

```
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".\model\Updated-Xception-diabetic-retinopathy.h5")
app = Flask(__name__)
# Authenticate using an IAM API key
client = Cloudant.iam('5e183e68-6288-4c71-84fb-5ce6fe11e2a7-bluemix',
                      'rn2GnKgXfP0v_qWogmV-MqgMlqkxeWHD9MTbjWdPWCJ1', 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 prediction():
    return render_template('prediction.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)
```

Feature 2

```
xception = Xception(input_shape=imageSize + [3], weights='imagenet', include_top=Fal
se)
# don't train existing weights for layer in xception.layers:
for layer in xception.layers:
  layer.trainable = False
# our Layers (- you can add more if you want x = Flatten(/(xception.output)
x = Flatten()(xception.output)

prediction = Dense(5, activation='softmax')(x)
model = Model(inputs=xception.input, outputs=prediction)

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

```
r=model.fit_generator(training_set,validation_data=test_set,epochs=30,steps_per_epoc
h=len(training_set)//32,validation_steps=len(test_set)//32)
```

APPENDIX:

```
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".\model\Updated-Xception-diabetic-retinopathy.h5")
app = Flask(__name__)
# Authenticate using an IAM API key
client = Cloudant.iam('5e183e68-6288-4c71-84fb-5ce6fe11e2a7-bluemix',
                      'rn2GnKgXfP0v_qWogmV-MqgMlqkxeWHD9MTbjWdPWCJ1', 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 prediction():
    return render_template('prediction.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)
! unzip diabetic-retinopathy-level-detection.zip
imageSize = [299,299]
trainPath = r"/content/preprocessed dataset/preprocessed dataset/training"
testPath = r"/content/preprocessed dataset/preprocessed dataset/testing"
! pip install kaggle
! mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
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
train_datagen = ImageDataGenerator (rescale = 1./255,
                                    shear_range = 0.2,
                                    zoom_range = 0.2,
                                    horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('/content/preprocessed dataset/prep
rocessed dataset/training',
                                                 target_size = (299, 299),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')
test_set = test_datagen.flow_from_directory('/content/preprocessed dataset/preproces
sed dataset/testing',
                                            target_size = (299, 299),
                                            batch_size = 32,
```

```
                                                class_mode = 'categorical')
xception = Xception(input_shape=imageSize + [3], weights='imagenet', include_top=Fal
se)
# don't train existing weights for layer in xception.layers:
for layer in xception.layers:
  layer.trainable = False
# our Layers (- you can add more if you want x = Flatten(/(xception.output)
x = Flatten()(xception.output)

prediction = Dense(5, activation='softmax')(x)
model = Model(inputs=xception.input, outputs=prediction)
model.summary()
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
r=model.fit_generator(training_set,validation_data=test_set,epochs=30,steps_per_epoc
h=len(training_set)//32,validation_steps=len(test_set)//32)
model.save('Updated-Xception-diabetic-retinopathy.h5')
```

GITUP LINK:

https://github.com/IBM-EPBL/IBM-Project-53568-1661418489

ADVANTAGES:

- Deep learning is well-suited for image analysis tasks. This isbecause deep learning algorithms can automatically learn featuresfrom images, which is essential for accurate image analysis.

- Deep learning is efficient at handling large amounts of data. This isimportant for medical image analysis, as medical images are often very large.

- Deep learning is scalable. This means that it can be used to train models on very large datasets, which is important for medicalimage analysis tasks where data is often limited.

- Deep learning is able to learn from data with little supervision.This is important for medical image analysis, as often there is limited labeled data available.

- Deep learning is robust. This means that it is less likely to over fit to the data, which is important for medical image analysis where data is often limited.

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 computationallyintensive, which can make them difficult to deploy in resource-limited settings.

CONCLUSION:

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

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.