HX8001
PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Project Report

Submitted by

Rahul M	**(113319106062)**
Padmanaban V	**(113319106052)**
Dhanush Shobin G	**(113319106018)**
Surendar R	**(113319106078)**
Siddagunta Meharnath Reddy	**(113319106072)**

Team ID : PNT2022TMID24488

BACHELOR OF ENGINEERING
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

VELAMMAL INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY: CHENNAI 600 025

Index

INTRODUCTION

Project Overview

Low cost Eye testing and efficient results, like computer aided testing. AI has been applied to image based medical field and suitable for processing complex image AI can assist diabetic retinopathy. Owing the characteristics of DL the training set have abundant and high quality images. Which will be easy and appropriate to evaluate. Quality of life of the diabetics with retinopathy can be improved and they can lead a life with color full vision. As the testing and analyzing is taken under low cost this can be developed into a business model so that it can help both the patient and the developer. 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

Purpose

Diabetic retinopathy is a complication of diabetes, caused by high blood sugar levels damaging the back of the eye (retina). It can cause blindness if left undiagnosed and untreated. However, it usually takes several years for diabetic retinopathy to reach a stage where it could threaten your sight. Diabetic eye screening is important as it helps to prevent sight loss. As someone with diabetes, your eyes are at risk of damage from diabetic retinopathy. Screening can detect the condition early before you notice any changes to your vision.

LITERATURE SURVEY

Existing problem

Diabetic Retinopathy (DR) is a common complication of diabetes mellitus, which causes lesions on the retina that effect vision. If it is not detected early, it can lead to blindness. Unfortunately, DR is not a reversible process, and treatment only sustains vision. DR early detection and treatment can significantly reduce the risk of vision loss. The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided

References

- M. I. Razzak, S. Naz and A. Zaib, "Deep learning for medical image processing: Overview challenges and the future" in Classification in BioApps, Springer, pp. 323-350, 2018.

- W. Zhang, X. Zhao, Y. Chen, J. Zhong and Z. Yi, "DeepUWF: An Automated UltraWide-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.

- A. Kwasigroch, B. Jarzembinski and M. Grochowski, "Deep CNN based decision support system for detection and assessing the stage of diabetic retinopathy", Proc. Int. Interdiscipl. PhD Workshop (IIPhDW), pp. 111-116, May 2018.

- 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.

- S. Seth and B. Agarwal, "A hybrid deep learning model for detecting diabetic retinopathy", J. Statist. Manage. Syst., vol. 21, no. 4, pp. 569-574, Jul. 2018.

- 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.

- L. Zhou, Y. Zhao, J. Yang, Q. Yu and X. Xu, "Deep multiple instance learning for automatic detection of diabetic retinopathy in retinal images", IET Image Process., vol. 12, no. 4, pp. 563-571, 2017.

- 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.

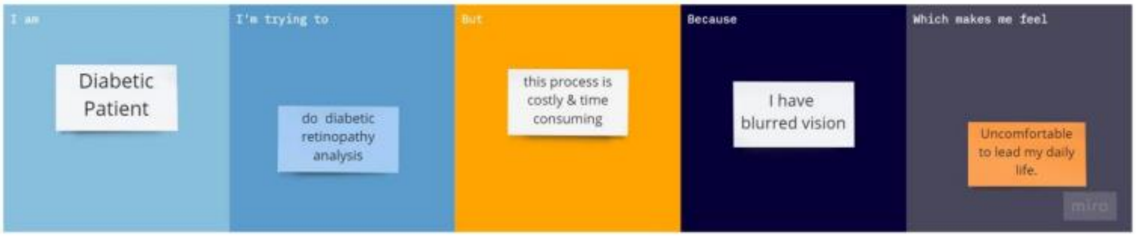
- S. Suriyal, C. Druzgalski and K. Gautam, "Mobile assisted diabetic retinopathy detection using deep neural network", Proc. Global Med. Eng. Phys. Exchanges/Pan Amer. Health Care Exchanges (GMEPE/PAHCE), pp. 1-4, Mar. 2018.

- S. Gupta, A. Panwar, A. Kapruwan, N. Chaube and M. Chauhan, "Real Time Analysis of Diabetic Retinopathy Lesions by Employing Deep Learning and Machine Learning Algorithms using Color Fundus Data," 2022 International Conference on Innovative Trends in Information Technology (ICITIIT), 2022, pp. 1-5, doi: 10.1109/ICITIIT54346.2022.9744228.

Problem Statement Definition

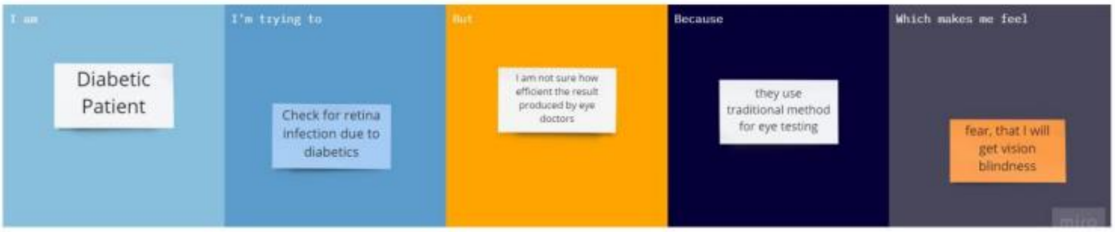
Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina). At first, diabetic retinopathy might cause no symptoms or only mild vision problems. But it can lead to blindness. The condition 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 1:-



I am	I'm trying to	But	Because	Which makes me feel
Diabetic Patient	do diabetic retinopathy analysis	this process is costly & time consuming	I have blurred vision	Uncomfortable to lead my daily life.

Patient 2:-

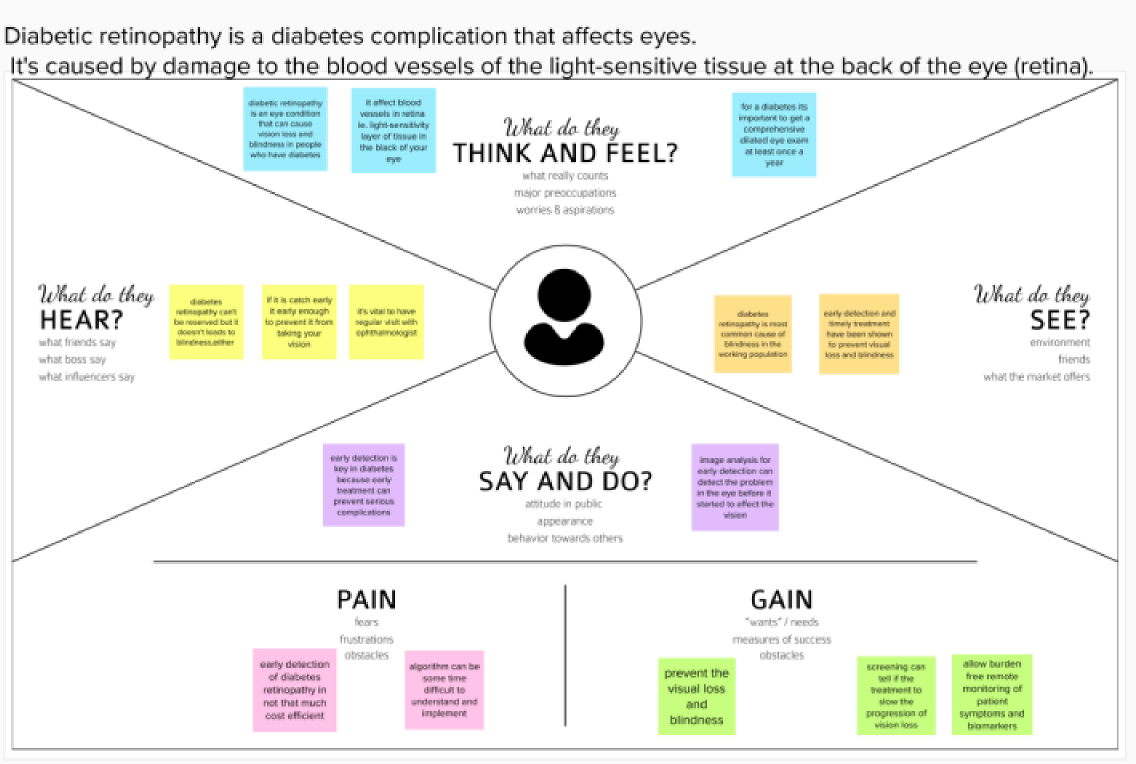


I am	I'm trying to	But	Because	Which makes me feel
Diabetic Patient	Check for retina infection due to diabetics	I am not sure how efficient the result produced by eye doctors	they use traditional method for eye testing	fear, that I will get vision blindness

Problem Statement (PS)	I am (Patient)	I'm trying to	But	Because	Which makes me feel
PS-1	Diabetic Patient	do diabetic retinopathy analysis	this process is costly & time consuming	I have blurred vision	Uncomfortable to lead my daily life.
PS-2	Diabetic Patient	Check for retina infection due to diabetics	I am not sure how efficient the result produced by eye doctors	they use traditional method for eye testing	fear, that I will get vision blindness

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



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).

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

diabetic retinopathy is an eye condition that can cause vision loss and blindness in people who have diabetes

it affect blood vessels in retina ie. light-sensitivity layer of tissue in the black of your eye

for a diabetes its important to get a comprehensive dilated eye exam at least once a year

What do they HEAR?
what friends say
what boss say
what influencers say

diabetes retinopathy can't be reserved but it doesn't leads to blindness,either

if it is catch early it early enough to prevent it from taking your vision

it's vital to have regular visit with ophthalmologist

What do they SEE?
environment
friends
what the market offers

diabetes retinopathy is most common cause of blindness in the working population

early detection and timely treatment have been shown to prevent visual loss and blindness

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

early detection is key in diabetes because early treatment can prevent serious complications

image analysis for early detection can detect the problem in the eye before it started to affect the vision

PAIN
fears
frustrations
obstacles

early detection of diabetes retinopathy in not that much cost efficient

algorithm can be some time difficult to understand and implement

GAIN
"wants" / needs
measures of success
obstacles

prevent the visual loss and blindness

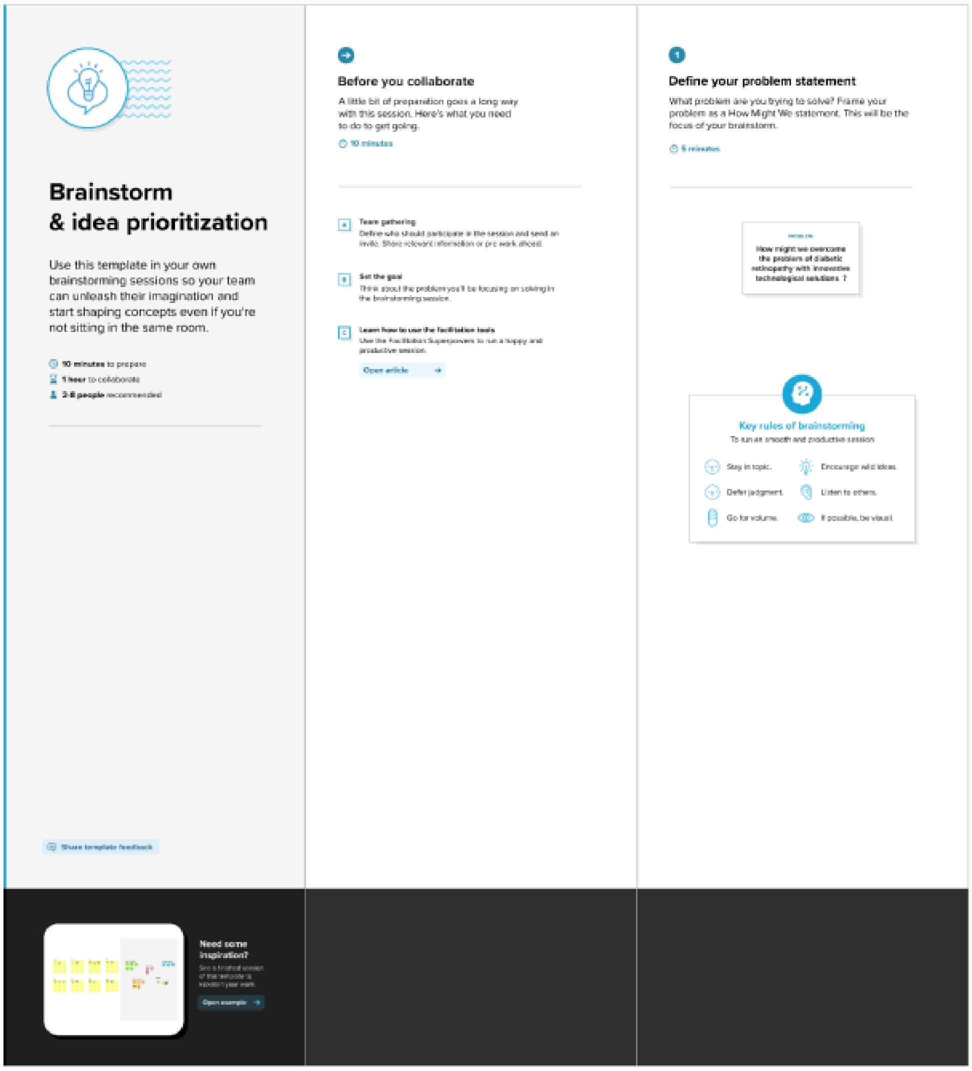
screening can tell if the treatment to slow the progression of vision loss

allow burden free remote monitoring of patient symptoms and biomarkers

Ideation & Brainstorming

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.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

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

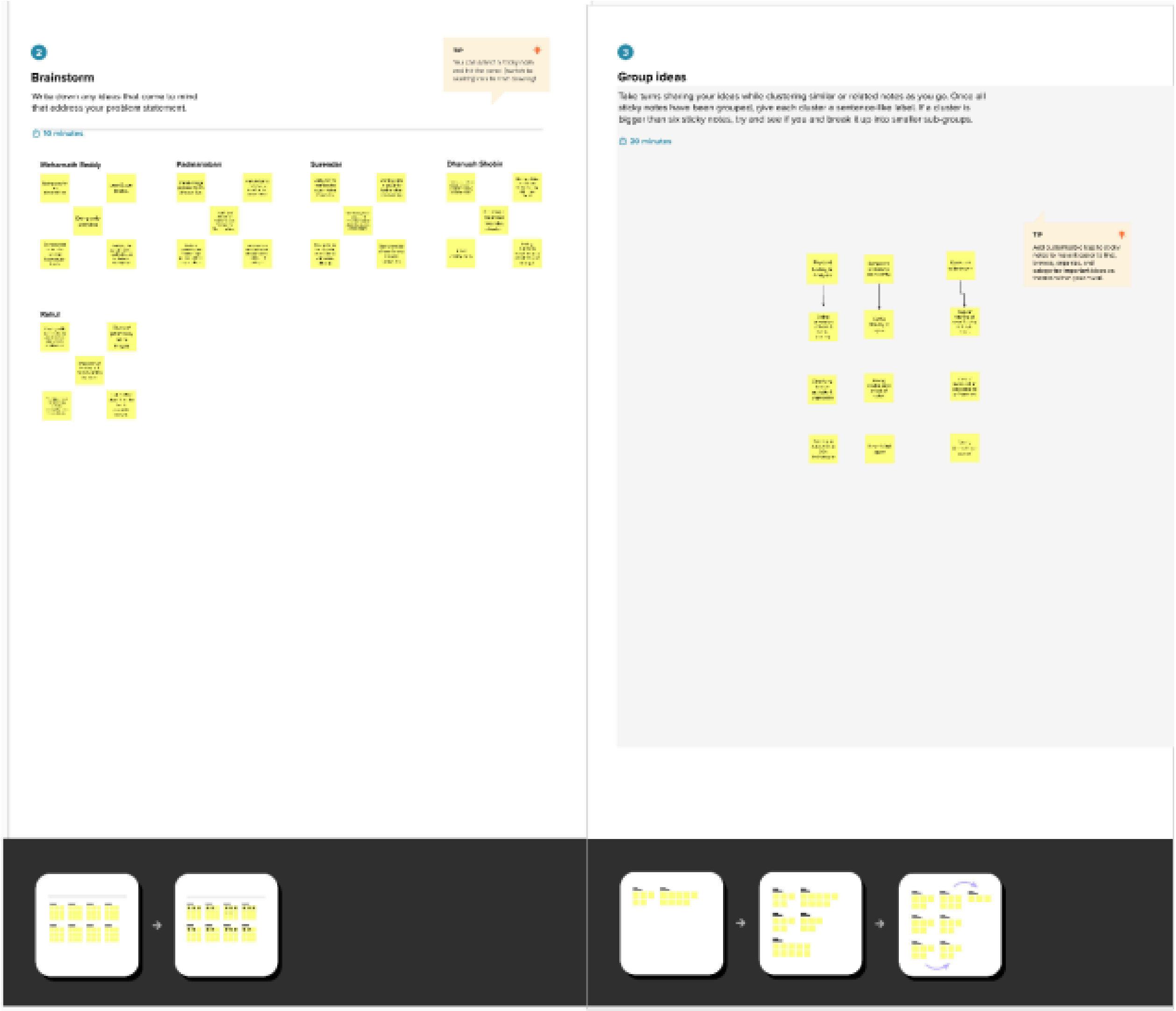
How might we overcome the problem of diabetic retinopathy with innovative technological solutions ?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

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

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

TIP

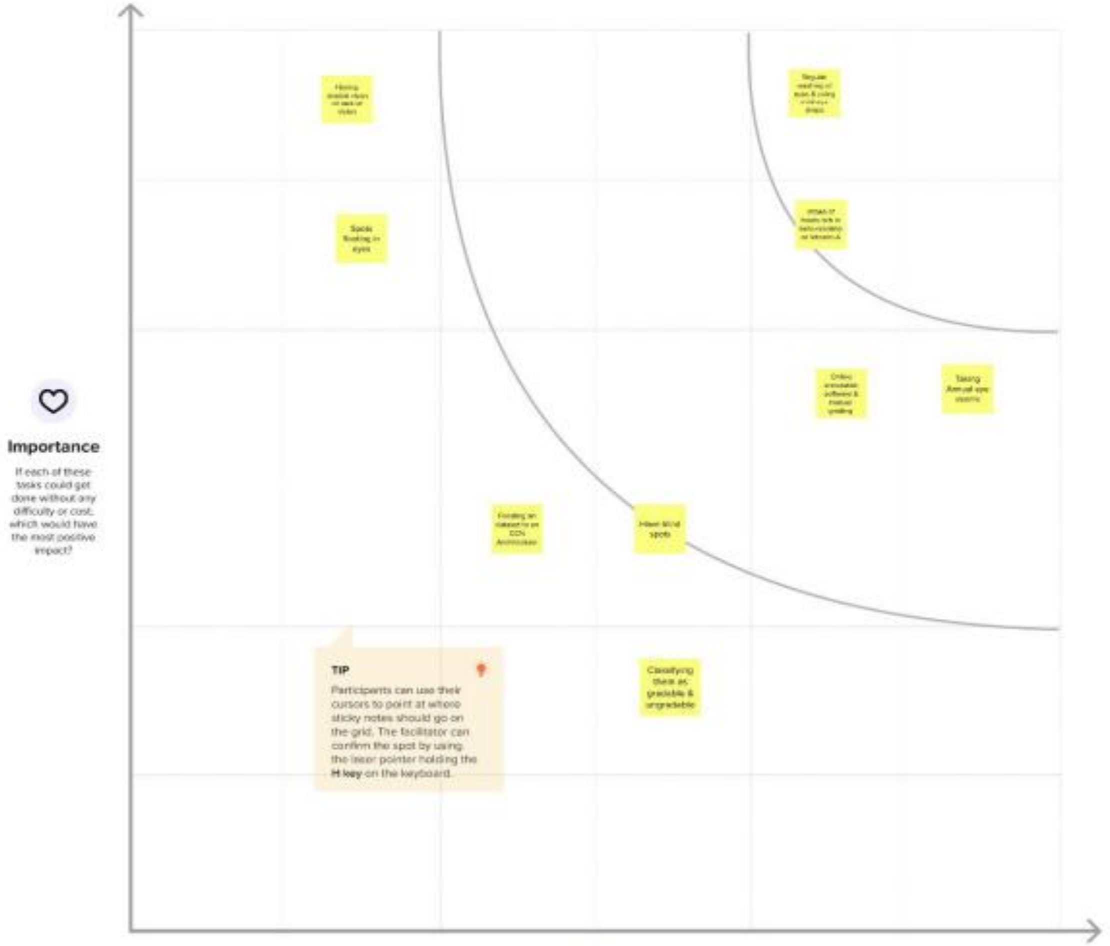
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

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


Having dental visits on each of visits
Spots floating in eyes
Regular washing of eyes & using [illegible] drops
Intake of healthy diet to [illegible] of Vitamin A
Online annotation software & manual grading
Taking Annual eye exams
Finding an dataset to an CCN Annotation
Have blind spots
Classifying them as gradable & ungradable

Importance

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

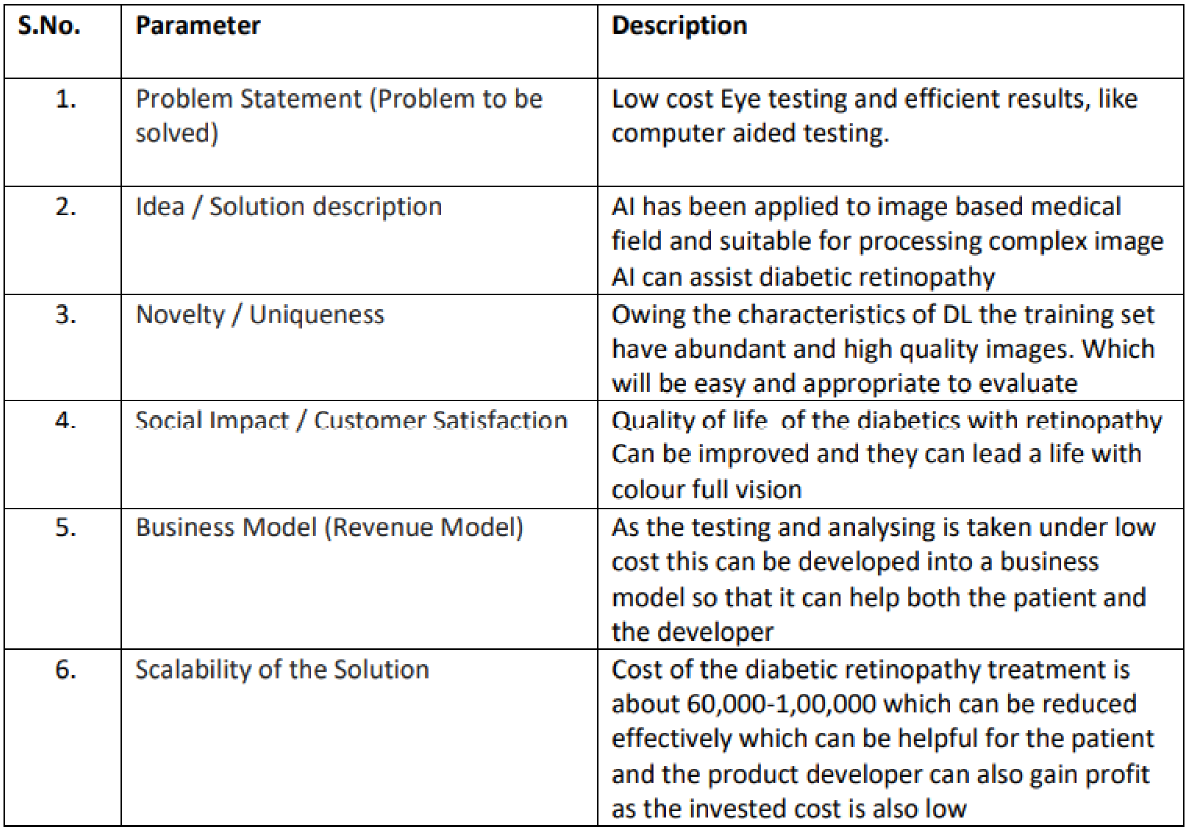
TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Feasibility

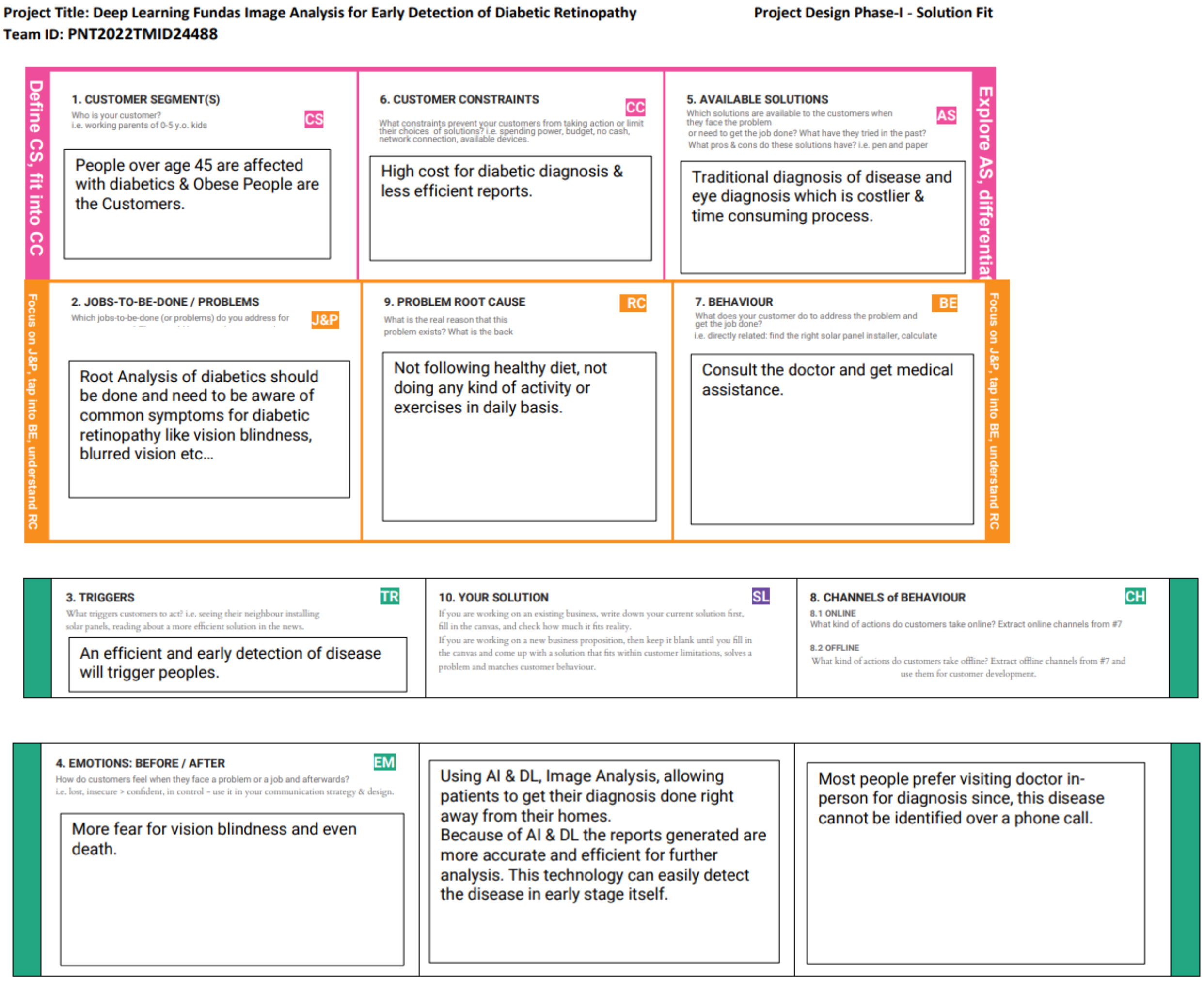
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

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

Solution fit



Project Title: Deep Learning Fundas Image Analysis for Early Detection of Diabetic Retinopathy **Project Design Phase-I - Solution Fit**

Team ID: PNT2022TMID24488

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

People over age 45 are affected with diabetics & Obese People are the Customers.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

High cost for diabetic diagnosis & less efficient reports.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past?
What pros & cons do these solutions have? i.e. pen and paper

Traditional diagnosis of disease and eye diagnosis which is costlier & time consuming process.

Explore AS, differentiat

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

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

Which jobs-to-be-done (or problems) do you address for

Root Analysis of diabetics should be done and need to be aware of common symptoms for diabetic retinopathy like vision blindness, blurred vision etc…

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back

Not following healthy diet, not doing any kind of activity or exercises in daily basis.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate

Consult the doctor and get medical assistance.

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

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

An efficient and early detection of disease will trigger peoples.

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

More fear for vision blindness and even death.

Using AI & DL, Image Analysis, allowing patients to get their diagnosis done right away from their homes.
Because of AI & DL the reports generated are more accurate and efficient for further analysis. This technology can easily detect the disease in early stage itself.

Most people prefer visiting doctor in-person for diagnosis since, this disease cannot be identified over a phone call.

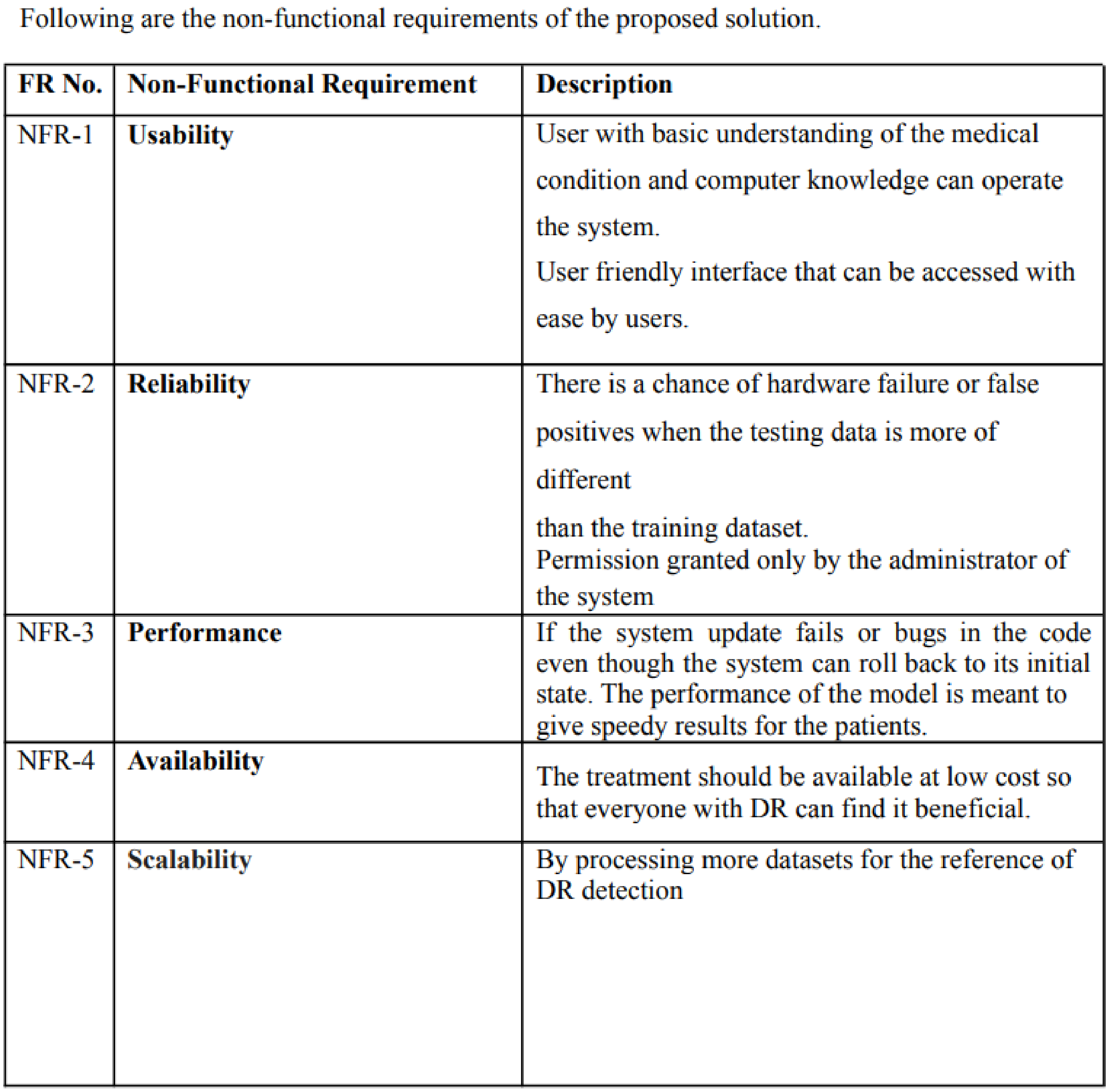
REQUIREMENT ANALYSIS

Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identify and selecting 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 true and false of the medical condition [Diabetic Retinopathy].
FR-4	Analysis	Based on the training the model should analyze the medical condition [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 [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



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

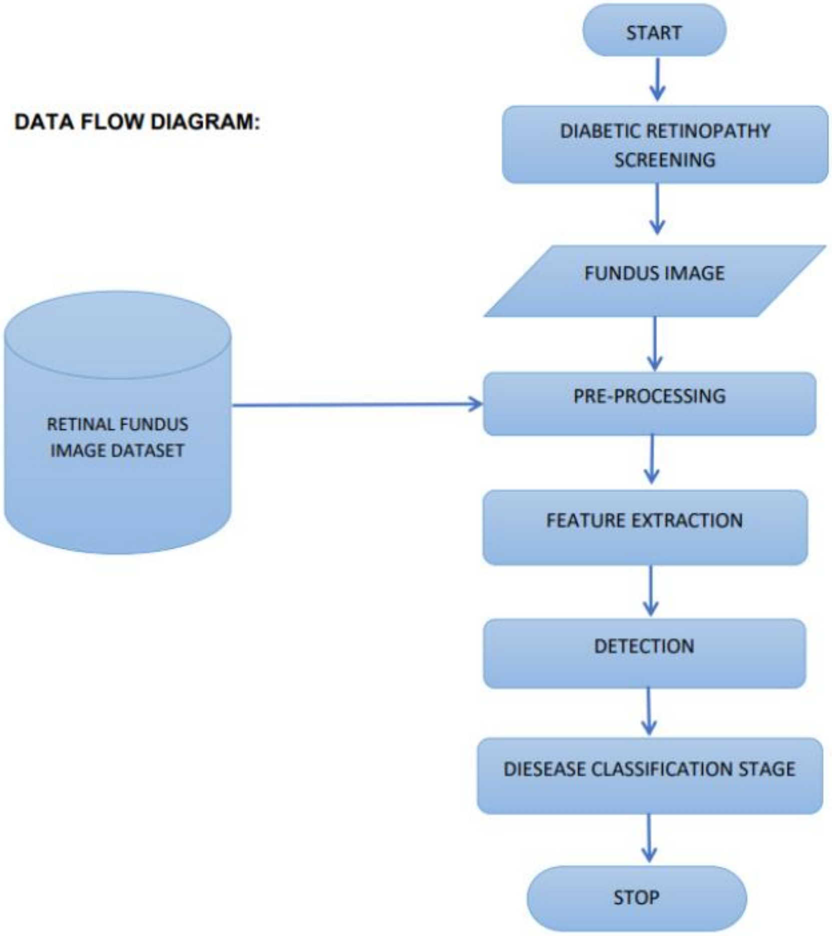
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User with basic understanding of the medical condition and computer knowledge can operate the system. User friendly interface that can be accessed with ease by users.
NFR-2	**Reliability**	There is a chance of hardware failure or false positives when the testing data is more of different than the training dataset. Permission 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

PROJECT DESIGN

Data Flow Diagrams

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.

DATA FLOW DIAGRAM:

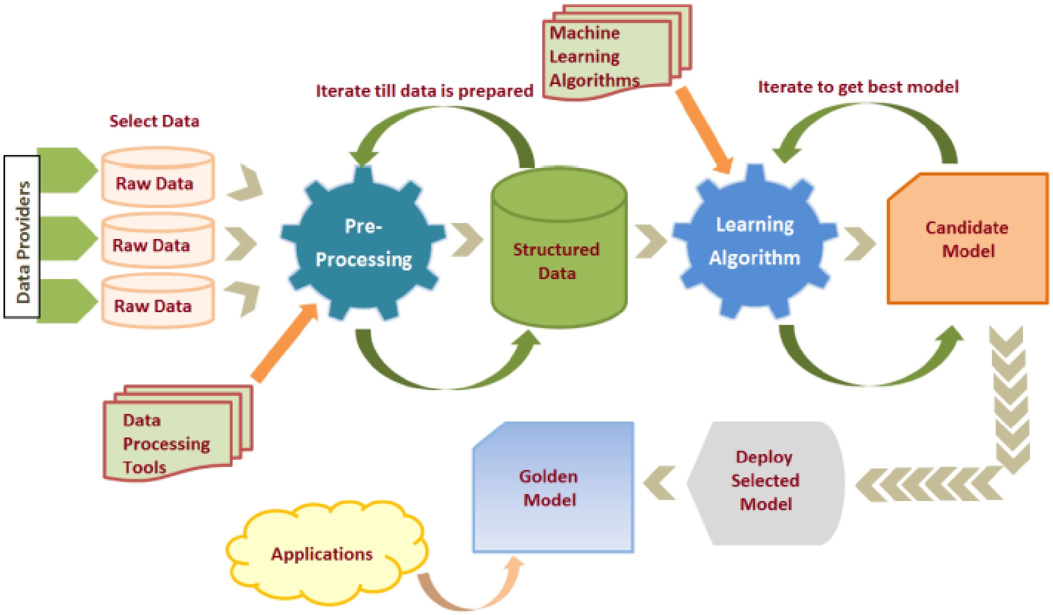

START
DIABETIC RETINOPATHY SCREENING
FUNDUS IMAGE
RETINAL FUNDUS IMAGE DATASET
PRE-PROCESSING
FEATURE EXTRACTION
DETECTION
DIESEASE CLASSIFICATION STAGE
STOP

Solution & 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

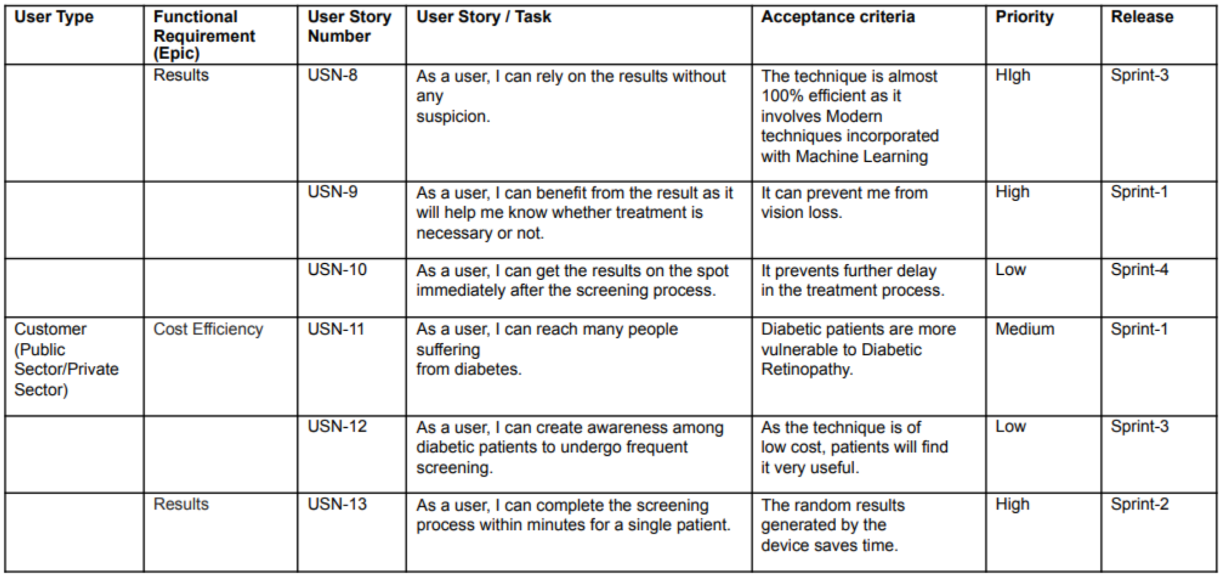
Solution Architecture Diagram:



User Stories

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

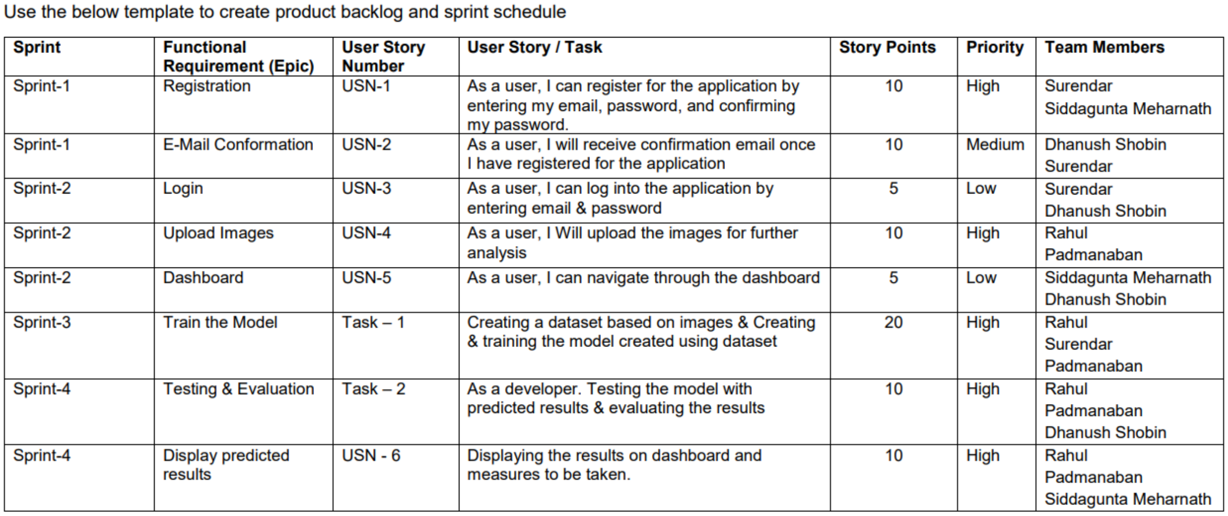
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can check whether I have Retinopathy or not by uploading the image of my eye by entering details.	I can upload or take image.	High	Sprint-1
	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 & SCHEDULING

Sprint Planning & Estimation



Use the below template to create product backlog and sprint schedule

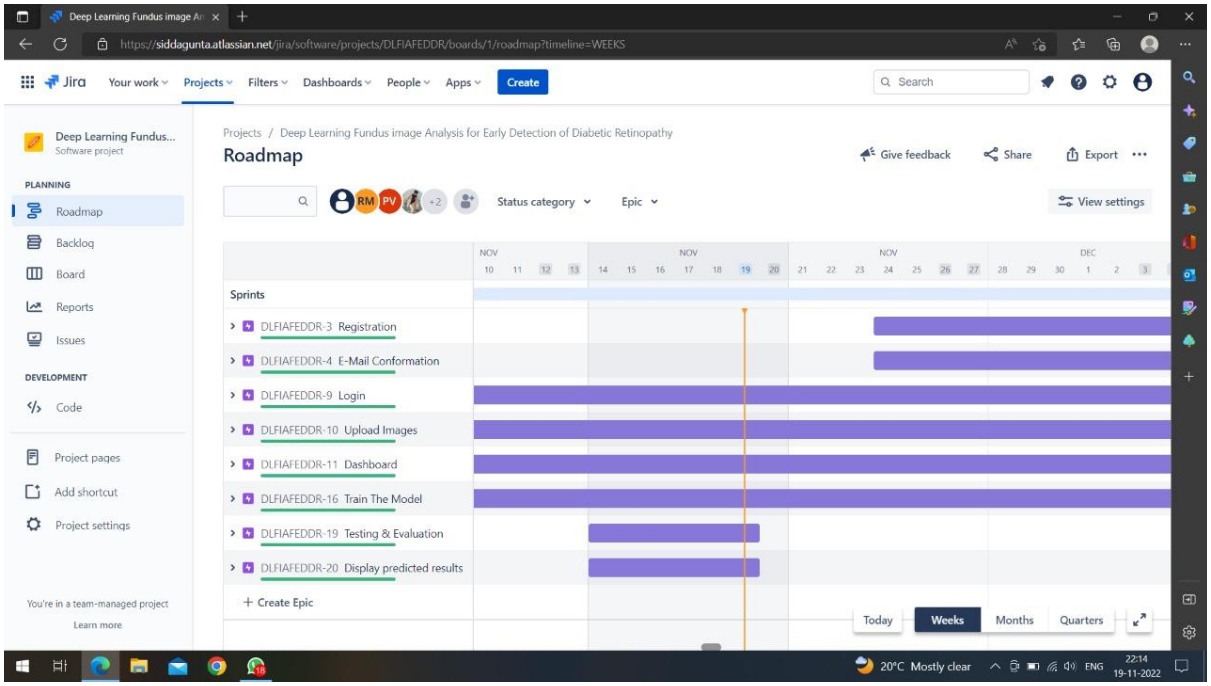
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Surendar Siddagunta Meharnath
Sprint-1	E-Mail Conformation	USN-2	As a user, I will receive confirmation email once I have registered for the application	10	Medium	Dhanush Shobin Surendar
Sprint-2	Login	USN-3	As a user, I can log into the application by entering email & password	5	Low	Surendar Dhanush Shobin
Sprint-2	Upload Images	USN-4	As a user, I Will upload the images for further analysis	10	High	Rahul Padmanaban
Sprint-2	Dashboard	USN-5	As a user, I can navigate through the dashboard	5	Low	Siddagunta Meharnath Dhanush Shobin
Sprint-3	Train the Model	Task – 1	Creating a dataset based on images & Creating & training the model created using dataset	20	High	Rahul Surendar Padmanaban
Sprint-4	Testing & Evaluation	Task – 2	As a developer. Testing the model with predicted results & evaluating the results	10	High	Rahul Padmanaban Dhanush Shobin
Sprint-4	Display predicted results	USN - 6	Displaying the results on dashboard and measures to be taken.	10	High	Rahul Padmanaban Siddagunta Meharnath

Sprint Delivery Schedule

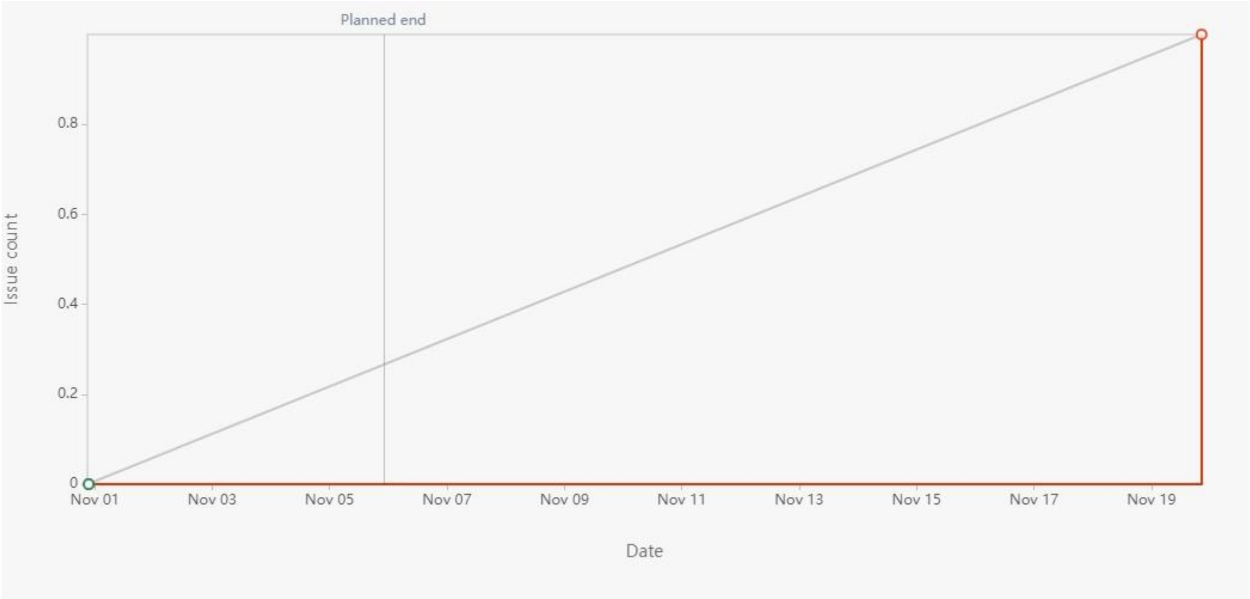
Project Tracker, Velocity & Burndown Chart: (4 Marks)

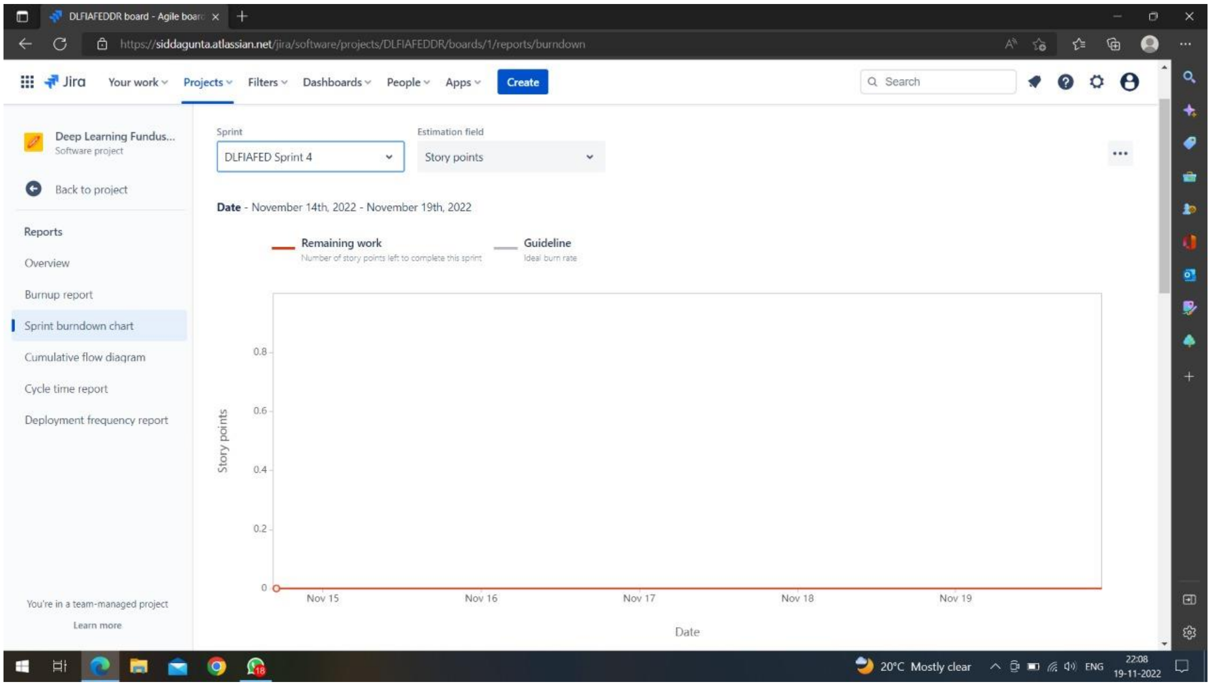
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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



DLFIAFEDDR-3 Registration
DLFIAFEDDR-4 E-Mail Conformation
DLFIAFEDDR-9 Login
DLFIAFEDDR-10 Upload Images
DLFIAFEDDR-11 Dashboard
DLFIAFEDDR-16 Train The Model
DLFIAFEDDR-19 Testing & Evaluation
DLFIAFEDDR-20 Display predicted results





Sprint: DLFIAFED Sprint 4

Estimation field: Story points

Date - November 14th, 2022 - November 19th, 2022

Remaining work — Number of story points left to complete this sprint

Guideline — Ideal burn rate

CODING & SOLUTIONING (Explain the features added in the project along with code)

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")
```

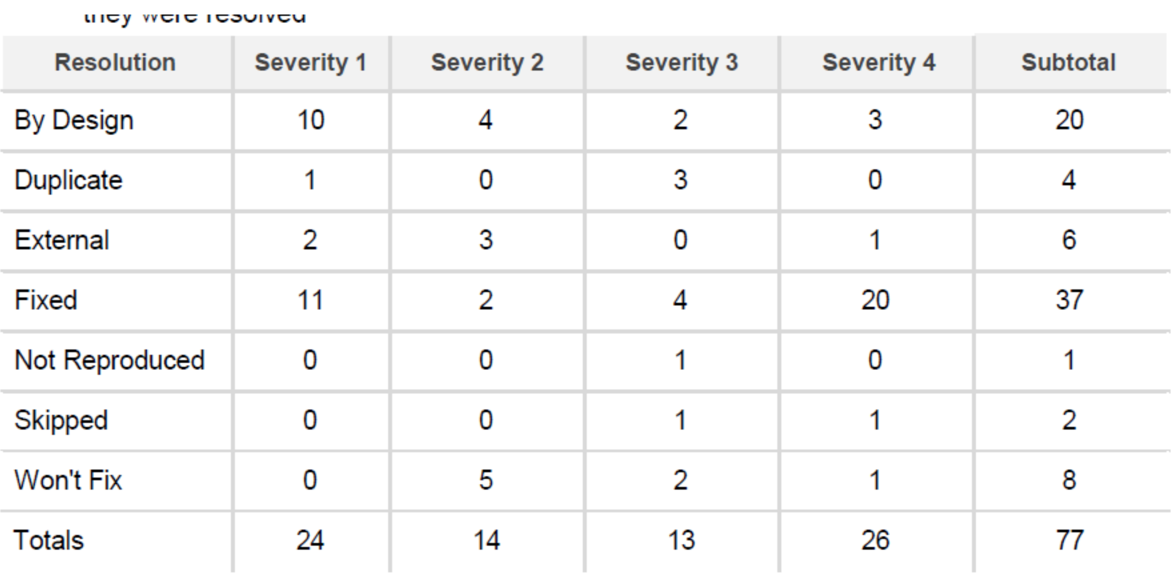
```
@app.route('/result', methods=["GET", "POST"])
def res():
    if request.method == "POST":
        f = request.files['image']
        # getting the current path 1.e where app.py is present
        basepath = os.path.dirname(__file__)
        # print ( " current path " , basepath )
        # from 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', inclu
de_top=False)
# 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=['a
ccuracy'])

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

TESTING

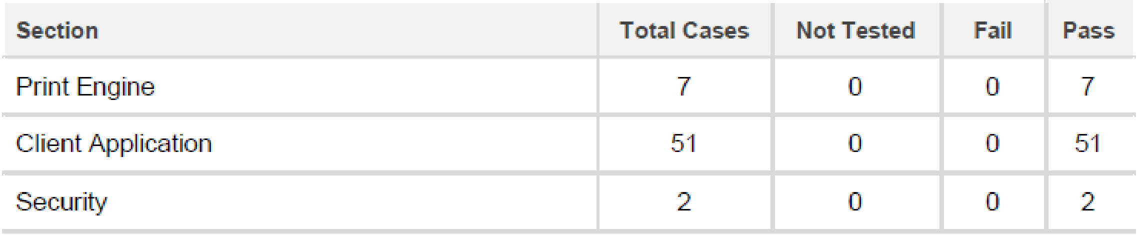
Test Cases

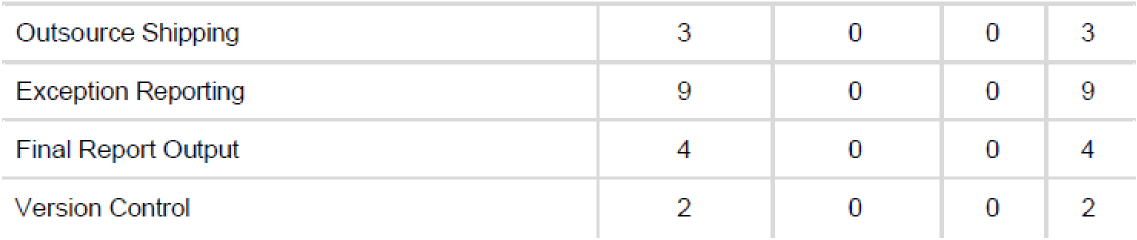


they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

User Acceptance Testing

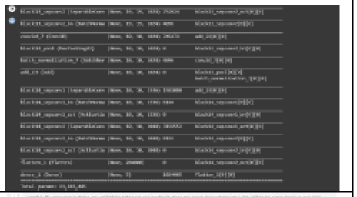
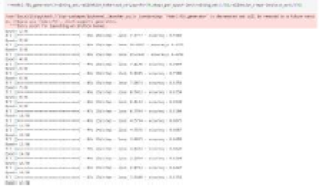




Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

RESULTS

Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 21,885,485 Trainable params: 1,024,005 Non-trainable params: 20,861,480	 
2.	Accuracy	Training Accuracy - 0.6354 Validation Accuracy – 3.6400	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	------

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.

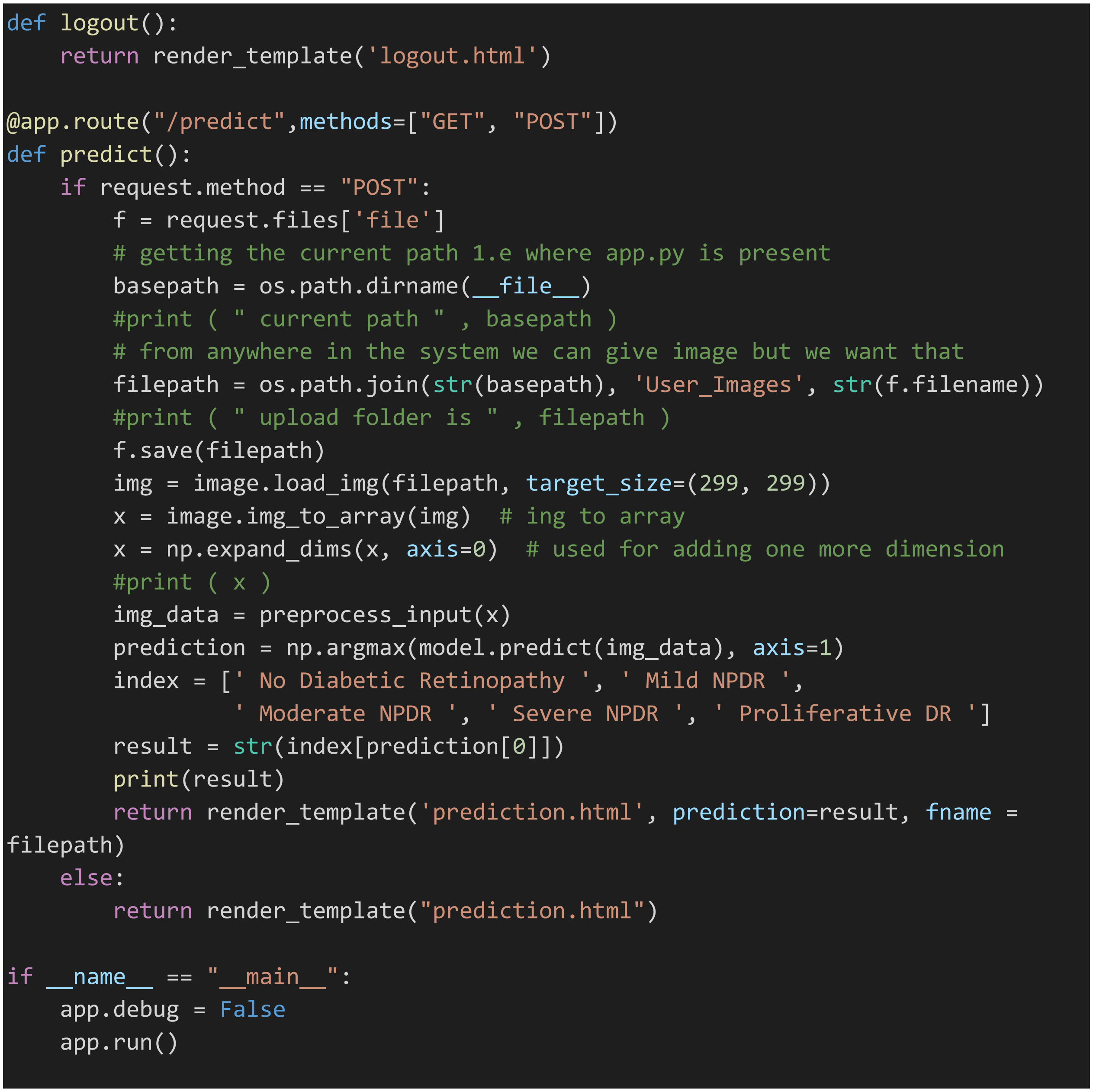
FUTURE SCOPE

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.

APPENDIX

Source Code

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
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('5e183e68-6288-4c71-84fb-5ce6fe11e2a7-
bluemix','rn2GnKgXfP0v_qWogmV-MqgMlqkxeWHD9MTbjWdPWCJ1',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","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 == "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')
```



```python
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)
        return render_template('prediction.html', prediction=result, fname =
filepath)
    else:
        return render_template("prediction.html")

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

ModelRetraining.ipynb

```
! pip install kaggle
! mkdir ~/.kaggle
! cp kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
```

In [ ]:

```
! kaggle datasets download arbethi/diabetic-retinopathy-level-detection/download?datasetVersionNumber=3
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
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
```

In [ ]:

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

In [ ]:

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

In [ ]:

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

In [ ]:

```
model.summary()
```

In [ ]:

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

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)
```

In [ ]:

```
model.save('Updated-Xception-diabetic-retinopathy.h5')
```

GitHub & Project Demo Link

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-53540-1661415224

Demo Link:

https://drive.google.com/file/d/1cwedkaid2IztCuHzUd6kwb0y2NrqVLNh/view?usp=share_link