Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team ID: PNT2022TMID31767

Submitted By

KIRTHIK A

NOEL MATHEW CHERIAN

SAKTHIVEL S

VENKATESH M

INDEX

INTRODUCTION:

1.1 PROJECT OVERVIEW:

Diabetes can have visible microvascular effects on the retina of the human eye, including diabetic retinopathy and macular edema, whose pictures are used for manual disease screening and diagnosis. Deep learning-based automated detection might be quite useful for this labour-intensive activity. We offer a deep learning system that identifies referable diabetic retinopathy equivalent to or better than reported in the earlier research, despite the fact that we only use a small part of photographs for the four-grade diabetic macular edema scales. These results suggest that a deep learning system might give performance above necessary levels while improving screening and diagnostic cost-effectiveness, and that the technique could be applied in clinical tests requiring finer grading. 2.6 million people were blind and visually handicapped.

1.2 PURPOSE:

Fully automated screening methods are necessary to prevent avoidable diabetic retinopathy-related blindness in the vast and expanding population of diabetics (DR). An automated, extremely accurate, cloud-based DR screening technology is offered by EyeArt and is capable of swiftly and correctly screening millions of images. This makes it possible to implement screening effectively and widely to help with the triage of DR patients who most urgently need eye therapy. The main goals of this course are to introduce students to the fundamental ideas and methods used in the processing of medical images and to pique their interest in further research and study in the field.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

Scar tissue can pull the retina away from the back of the eye as a result of the aberrant blood vessels linked to diabetic retinopathy. This may result in bright flashes, floating patches in your field of vision, or significant vision loss.

2.2 REFERENCES:

[1] V.V.K. Raju, S.Sagar, B.Divya, K.Madhulika, D.Sandeep “Retinal Image Analysis for Diabetic Retinopathy”, The international journal of analytical and experimental modal analysis, May 2020.

[2] Sayres, R. et al. Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy.

opthamology 126, 552–564 (2019).
[3] Sahlsten, J. et al. Deep learning fundus image analysis for diabetic retinopathy and macular edema grading. Sci. Rep. 9, 1–11 (2019).
[4] Gulshan, V. et al. Performance of a deep-learning algorithm vs manual grading for detecting diabetic retinopathy in India. JAMA Ophthalmol. 137, 987– 993 (2019).
[5] Nagasawa, T. et al. Accuracy of ultrawide-field fundus ophthalmoscopy assisted deep learning for detecting treatment-naïve proliferative diabetic retinopathy. Int. Ophthalmol. 39, 2153–2159 (2019).
[6] Ali A, Qadri S, Mashwani WK, Kumam W, Kumam P, Naeem S, Goktas A, Jamal F, Chesneau C, Anam S, Sulaiman M (2020) Machine learning based automated segmentation and hybrid feature analysis for diabetic retinopathy classification using fundus image.
[7] Al Youbi WL, Shalash WM, Abulkhair MF (2020) Diabetic retinopathy detection through deep learning techniques: a review. Inf Med Unlocked 20:100377
[8] Cano J, O'neill WD, Penn RD, Blair NP, Kashani AH, Ameri H, Kaloostian CL, Shahidi M (2020) Classification of advanced and early stages of diabetic retinopathy from non-diabetic subjects by an ordinary least squares modelling method applied to OCTA images. Biomed Opt Express 11(8):4666–4678
[9] Bhardwaj C, Jain S, Sood M (2021) Hierarchical severity grade classification of non-proliferative diabetic retinopathy. J Ambient Intell Humaniz Comput 12:2649–2670
[10] Chaki J, Ganesh ST, Cidham SK, Theertan SA (2020) Machine learning and artificial intelligence based diabetes mellitus detection and self-management: a systematic review. J King Saud Univ Comput Inf Sci.

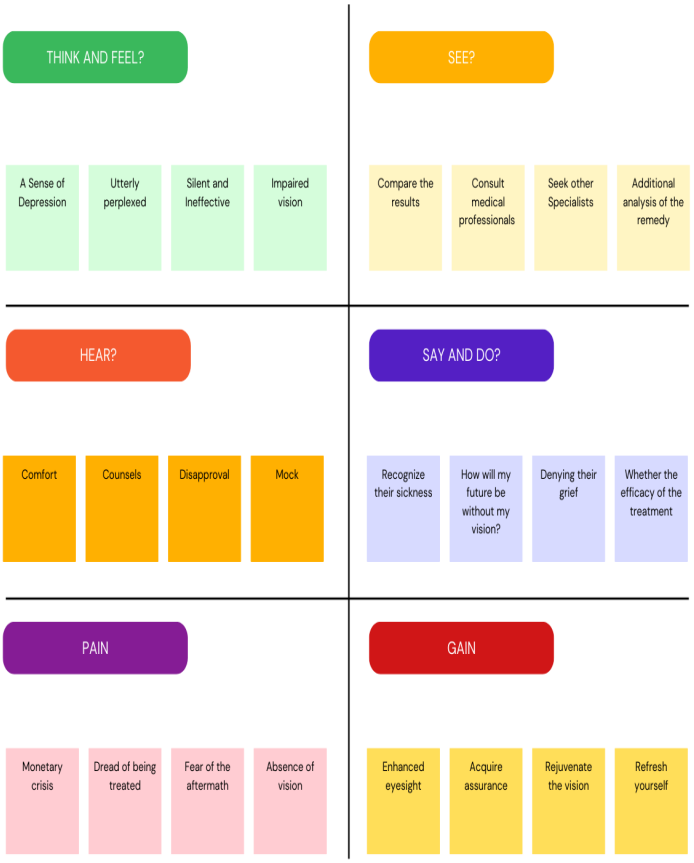
2.3 PROBLEM STATEMENT DEFINITION:

Diabetic retinopathy (DR), which causes lesions on the retina that impair vision, is a common complication of diabetes mellitus. If it is not detected in time, blindness might follow. If such an algorithm were to be built on larger and more varied datasets, it may assist early identification, referral to a retina expert for more frequent monitoring, and perhaps consideration of early intervention. Furthermore, it could increase the number of patients who join in therapy studies that focus on DR. If you have type 1 or type 2 diabetes, a disease known as diabetic retinopathy (DR) may appear in your eyes. The tissue that lines the

back of the eye, known as the retina, is damaged as a result. One can see thanks to signals that are sent from the retina to the brain. Diabetes results in high blood sugar levels, which damage the body's blood vessels. One may first have partial vision loss as a result, and eventually lose all of their vision. The good news is that additional vision loss can be avoided with appropriate therapy when a patient's symptoms are identified in their early stages. Therefore, it is crucial that we identify diabetic retinopathy in its early stages.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION AND BRAINSTORMING:

IDEATION 1:

This project focuses on methods for detecting the first instances of diabetes in people using digital image processing. It analyses the input image of the eye that was captured and determines whether the retina of the subject's eye is showing any signs of diabetes. This information is then used to produce a balanced and reliable bioinformatics report that aids in the early detection of the condition.

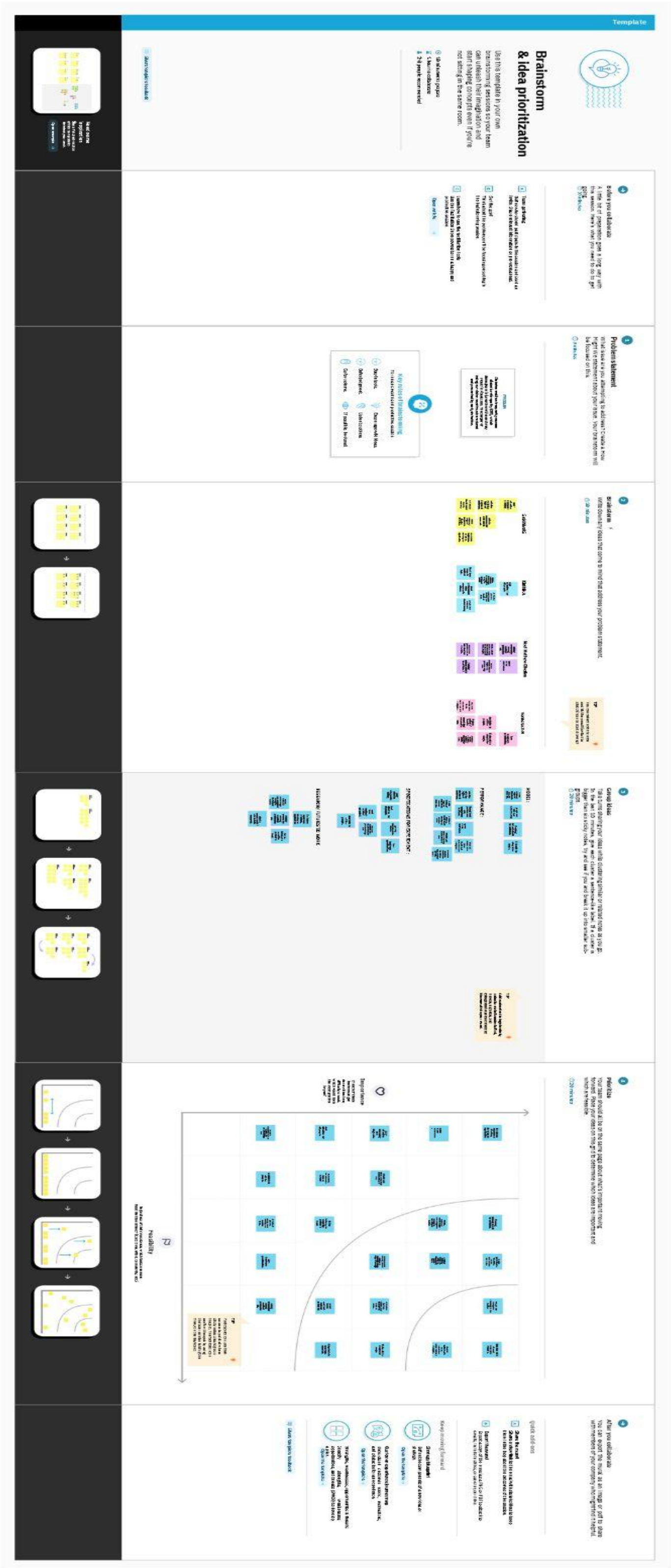
IDEATION 2:

The study, which is based on the deteriorating situation in the developing world, contends that diabetic retinopathy, a primary cause of blindness, may soon become a significant clinical issue. Therefore, it is crucial to identify diabetic retinopathy. This study utilises the Raspberry Pi kit to analyse retinal pictures to determine if they are normal or abnormal and to determine the metrics of DR. Python's Threshold and the Color-k clustering, water, mean shift, and distance algorithms are used to identify diabetic retinopathy in retinal pictures.

IDEATION 3:

The retinal microvasculature, which may develop as a result of diabetes mellitus, is what causes diabetic retinopathy. Unchecked and severe cases of diabetic retinopathy can lead to blindness. It takes a lot of time and effort to manually review fundus pictures to look for morphological changes in microaneurysms, exudates, blood vessels, haemorrhages, and the macula. Inter variability for the observer and computer-aided systems can make it simple. In order to identify non-proliferative diabetic retinopathy, numerous procedures for spotting microaneurysms, haemorrhages, and exudates are reviewed in this work. Techniques for detecting blood vessels are also covered for proliferative diabetic retinopathy diagnosis.

BRAINSTORMING:



3.3 PROPOSED SOLUTION:

Numerous AI-based methods have been put forth by the scientific community for the classification and diagnosis of diabetic retinopathy using fundus retinal pictures. As a deep learning approach, this takes into account a convolutional neural network (CNN), which is employed for the early identification of diabetic retinopathy. Regular dilated eye exams are the most effective way to detect and treat diabetic retinopathy, which threatens vision. They can help prevent blindness and are reasonably priced. They have their needs met via this service at no cost to them. Because it prevents the sick patient from getting blind, this may be employed as a commercial strategy. The vast majority of everyday citizens and hospitals will utilise this service. This programme will be able to grow the necessary medical diagnosis and guidance (integrated with their everyday lifestyle)

Project Design Phase-I

Proposed Solution Template

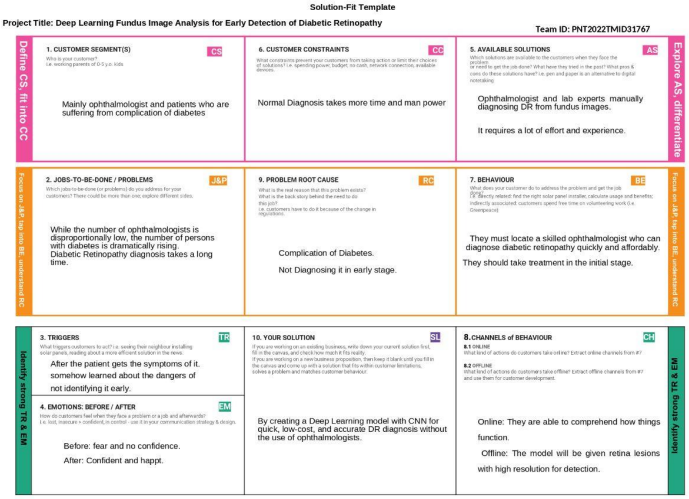
Date	15 October 2022
Team ID	PNT2022TMID31767
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	2 Marks

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Fundus image analysis for early cancer detection • diabetic retinal disease. • Analyze the DR level • To determine whether or not DR is present
2.	Idea / Solution description	The plan is to use the fundus to identify diabetic retinopathy. as soon as possible to ensure that People and patients may go to their necessary medical care and eyesight prevention degradation or loss of vision for all time. Since this DR cannot be completely cured, we will create a deep learning model. (CNN) to accurately identify DR and people at risk of losing their vision should be saved.
3.	Novelty / Uniqueness	On the basis of the level of DR performed during analysis, a class-based classifier will be provided. As part of the effort, we'll also test out a transfer learning strategy that has the potential to be very successful and lead to improved performance.

4.	Social Impact / Customer Satisfaction	People who lose their vision could actually benefit from this and live. Early analysis and detection of DR is vital for minimising societal impact because it can help patients keep their eyesight.
5.	Business Model (Revenue Model)	Doctors can analyse and identify DR using this model, in which it functions as a service model for public hospitals and a business model for private hospitals. Even it can be used as a business model by being exported to other nations that require it.
6.	Scalability of the Solution	There are more and more approaches to scale the solution so that the model is simple to combine with emerging technologies.

3.4 PROBLEM SOLUTION FIT:



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENT:

Project Design Phase-II
Solution Requirements (Functional & Non-functional)

Date	22 October 2022
Team ID	PNT2022TMID31767
Project Name	Diabetic Retinopathy
Maximum Marks	4 Marks

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR NO:	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identifying and selecting dataset	To improve the model's performance, the right dataset must be chosen.
FR-2	Invitation and information	Invite the whole cohort to the screening, and provide information that is suitable for each group. to make participation with knowledge possible.
FR-3	Training	The libraries required for the model's training must be imported.
FR-4	Diagnosis	Determine the false positives and diagnose real cases.
FR-5	Testing	To determine if the model is properly trained to forecast the medical condition, run screening tests on several sets of data.
FR-6	Reporting	Report the results to find false negatives and boost the screening program's efficacy.
FR-7	Intervention/Treatment/Followup	Testing the model enables us to choose the best course of action.

Non-Functional Requirements:

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

FR NO:	Non-Functional Requirement	Description
FR-1	Usability	The system may be used by anybody with a basic awareness of the medical condition and computing skills. User-friendly interface that is simple for people to access.

FR-2	Security	Deep learning AI can be more accurate when detecting sensitive organs and tissues while also minimising blood loss, infection risk, and discomfort.
NFR-3	Reliability	When the testing data is more disparate, there is a possibility of hardware malfunction or false positives.
NFR-4	Performance	Even if the system may rollback to its original state if a system upgrade fails or there are faults in the programming. The model's performance is intended to provide patients with quick results.
NFR-5	Availability	The treatment should be available at low cost so that everyone with DR can find it beneficial.
NFR-6	Scalability	processing more datasets for DR detection reference.

5. PROJECT DESIGN

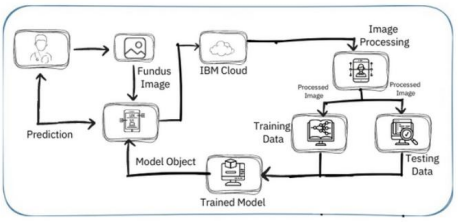
5.1 DATA FLOW DIAGRAMS AND USER STORIES:

Project Design Phase-II
Data Flow Diagram & User Stories

Date	03 October 2022
Team ID	PNT2022TMID36023
Project Name	Project - 2665-1658480813
Maximum Marks	4 Marks

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.



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Login	USN-5	As a user, I can log onto the application by entering email & password	I can login the application by email and access the dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can navigate through sections of the applications	I can navigate the sections of the applications	High	Sprint-2
Administrator	Login	USN-1	As a administrator, I can log into the application by entering email and password	I can Login the application by email and access the dashboard	High	Sprint-1
	Dashboard	USN-2	As an administrator, I can navigate through various sections of the application	I can navigate the sections of the application	High	Sprint-2
		USN-3	As an administrator, I can update data in database	I can update all data in the database.	High	Sprint-3
		USN-4	As an administrator, I can view all stages of the product	I can view and review all stages of the product	High	Sprint-3
		USN-5	As an administrator, I can add or remove user	I can add or remove an user from the application	High	Sprint-3

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 register for the application by entering my email, password and confirming my password.	I can access my account / dashboard	HIgh	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register& access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can login the application by email and access the dashboard	High	Sprint-1
	Dashboard	USN-6	As a user, I can navigate through various sections of the application.	I can navigate the sections of the application	High	Sprint-2
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering, password, and confirming my password	I can access my account/ dashboard	High	Sprint-1
Customer Care Executive		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can review confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register& access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register and access the dashboard with Gmail Login	Medium	Sprint-1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	22-10-2022
Team ID	PNT2022TMID31767
Project Name	Deep Learning Fundus Image Analysis for early detection of Diabetic Retinopathy
Maximum Marks	4 Marks

Architecture:

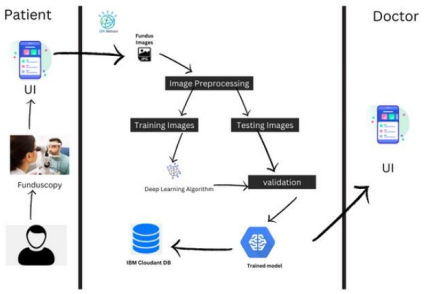


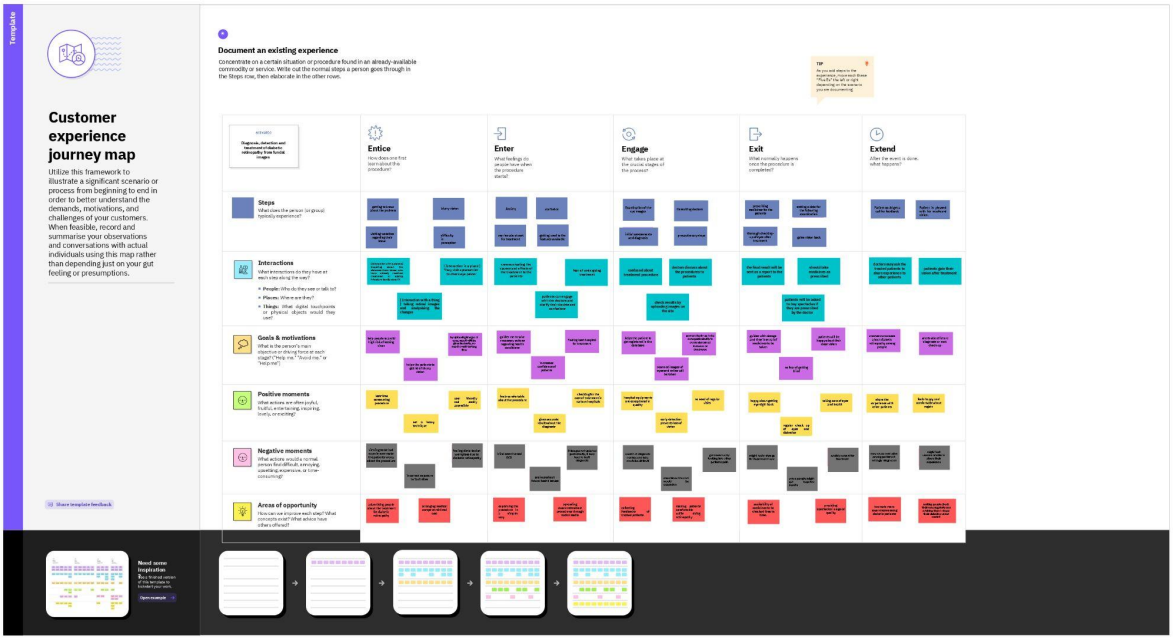
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	How the user interacts with the application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript, Bootstrap
2	Application Logic-1(Back-end)	Logic for a process in the application and the UI	Flask -Python
3	Application Logic-2	Logic for a process in the application and cloud	IBM Watson STT service
4	Cloud Database	Database Service on Cloud	IBM-Cloud.
5	File Storage	File storage requirements	IBM-Block Storage
6	Machine Learning Model	Purpose of Machine Learning Model	Deep Learning, Convolutional Neural Network.
7	Infrastructure (Server / Cloud)	Application Deployment in Cloud	IBM Cloud.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	List the open-source frameworks used	Tensorflow, Scikit-learn.
2	Security Implementations	Email constraint and database	Mail notifications, IBM free trial cloud, instant services.
3	Scalable Architecture	Scalability of architecture (3 – tier, Micro-services) is extensible as it is web-oriented with database application	3-tier, Cloud DB.
4	Availability	Availability of application (e.g. use of load balancers, distributed servers etc.)	Load balancer
5	Performance	Session management & Model Accuracy	User sessions from the Python-based automatic controller. It depends upon the input images.

5.3 CUSTOMER JOURNEY:



6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION:

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	29 October 2022
Team ID	PNT2022TMID31767
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	4 Marks

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	26 Oct 2022	01 Nov 2022	20	01 Oct 2022
Sprint-2	20	6 Days	01 Oct 2022	07 Nov 2022	20	07 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

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

6.2 SPRINT DELIVERY AND SCHEDULE:

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Date	28 October 2022
Team ID	PNT2022TMID31767
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	4 Marks

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

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection	Task-1	To build a Deep learning Model which begins with the process of splitting data into training and testing set.	4	Medium	Sakthivel S & Kirthik A
Sprint-1	Data preprocessing	Task-2	We import the required libraries for preprocessing. We instantiate the ImageDataGenerator class to configure and augment different types of image data.	5	Low	Kirthik A
Sprint-1	Data Preprocessing	Task-3	Application of the ImgaeDataGenerator to the Train and Test Set.	7	Medium	Kirthik A
Sprint-1	Building Homepage	USN-1	As a user, she will be given a brief description in the homepage.	4	Low	Venkatesh M
Sprint-2	Feature Extraction	Task-4	Build a CNN Model and only use it as a feature extraction by freezing the convolution blocks.	8	High	Sakthivel S & Kirthik A
Sprint-2	Building the layers	Task-5	Adding of dense layers with the aid of Keras Addition of optimizers,choosing loss function and the metrics.	7	High	Venkatesh M & Noel Mathew Cherian
Sprint-2	Train, Save,Test	Task-6	To train the model with the configured neural Network and save the model.Test the built model	3	High	Sakthivel S & Noel Mathew Cherian
Sprint-2	Building Registration page	USN-2	As a user, she will be able to register for the application.	2	Low	Venkatesh M & Noel Mathew Cherian
Sprint-3	Create Service Instance	Task-7	Configure the location of resources, such as web server, and Cloud Storage for an application	7	High	Kirthik A & Venkatesh M
Sprint-3	Configuring credentials and creating DB	Task-8	Define the credentials that are required to access the services offered by IBM Cloudant and add users to access the DB.	6	High	Sakthivel S & Venkatesh M
Sprint-3	Create Tables in DB	Task-9	Structure the required tables with necessary attributes in Cloudant DB.	4	Medium	Kirthik A
Sprint-3	Building Login Page	USN-3	As a user, she will be able to login using her credentials.	3	Low	Venkatesh M & Noel Mathew Cherian
Sprint-4	Building prediction page	USN-4	As a user, she will be able to receive the diagnosis on her diabetic retinopathy.	2	Medium	Venkatesh M & Noel Mathew Cherian
Sprint-4	Building Logout Page	USN-5	As a user, she will be able to logout of her account in this page.	2	Medium	Venkatesh M & Noel Mathew Cherian
Sprint-4	Build python code	Task-9	Import the libraries and Initialise the necessary modules	1	Medium	Sakthivel S & Venkatesh M
Sprint-4		Task-10	Use the database using initiated client and rendering HTML pages	2	Medium	Venkatesh M
Sprint-4		Task-11	Configuring the registration, login pages and validating the credentials.	2	Medium	Venkatesh M

Sprint-4		Task-12	Showcasing the model's prediction on UI.	1	High	Sakthivel S
Sprint-4	Run the application.	Task-13	Run the application in the anaconda prompt to check the application.	2	High	Kirthik A & Noel Mathew Cherian
Sprint-4		Task-14	In the homepage, after logging on using credentials, upload the image to predict the diagnosis on diabetic retinopathy.	5	High	Sakthivel S
Sprint-4	Train Model On IBM	Task-15	train the model on IBM and integrate it with the flask Application.	3	High	Sakthivel S

7. CODING AND SOLUTIONING

We have created a website that verifies individuals and allows them to post and assess the severity of diabetes. We have created a multilayer deep convolutional neural network that can identify which eyes in a user's photograph have the condition that affects diabetics the most severely. When asked for a forecast, the programme will categorise the photos into 5 categories of diabetes. Additionally, we have created a texting service for diabetics to receive messages.

8. TESTING

8.1 TEST CASES AND USER ACCEPTANCE TESTING:

Acceptance Testing
UAT Execution & Report Submission

Date	16 November 2022
Team ID	PNT2022TMID31767
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy project at the time of the release to User Acceptance Testing (UAT).

2.Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	4	2	2	1
Fixed	0	2	2	2	8
Skipped	6	0	0	1	1
Won't Fix	0	0	0	1	2
Totals	0	6	4	6	1
	1				1
	6				3
					2

3.Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	1	0	0	1
Security	7	0	0	7
Exception Reporting	2	0	0	2
Final Report Output	2	0	0	2
Version Control	5	0	0	5
	1			1

9. RESULTS

9.1 PERFORMANCE METRICS:

MODEL SUMMARY:

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

ACCURACY:

```
loss: 2.9340 - accuracy: 0.7292
```

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

• Using fundus pictures to classify exudates, microaneurysms, and haemorrhages, digital image processing technologies can be used to diagnose diabetic retinopathy and assist in identifying the early stages of diabetes.

• For the automated identification of diabetic retinopathy, the diagnostic approaches, assessment and comparison approaches, facts, and procedures are suggested.

• The outcomes of automatic diabetic retinopathy techniques aid in identifying the condition in its early stages and allow for the implementation of preventative measures to lessen diabetic patients' losses.

• Massive efforts have been undertaken over the past few decades to lessen diabetes consequences, including diabetic retinopathy.

• Early diabetic retinopathy and diabetic macular edema may now be diagnosed with greater accuracy thanks to new diagnostic techniques including ultrawide field fundus fluorescein angiography and spectral domain.

• This aids in the early discovery of retinopathy; prompt treatment of this illness will avoid irreversible visual loss.

• Injections of anti-inflammatory drugs or vascular endothelial growth factor inhibitors may aid in the process of the new blood vessels contracting in proliferative diabetic retinopathy.

DISADVANTAGES:

• More study is needed in this area to reliably identify the early stages of diabetic symptoms in humans. Ongoing research will undoubtedly result in more reliable ways for identifying the early stages of diabetes, which will help save human lives.

• Anti-vascular endothelial growth factors are currently being utilised widely and with encouraging results to treat diabetic retinopathy and retinal edema.

• The long-term consequences and socioeconomic implications of these chemicals are still unknown.

• Even If a person with diabetes has appropriate eye care on a regular basis and medication when necessary, DR will occasionally result in total blindness.

• Early signs of DR are present in nonproliferative diabetic retinopathy, and it is crucial to identify and assess DR when it first appears.

11. CONCLUSION

Techniques for image segmentation work well and are equivalent to approaches used in practice. Intensity, texture, and picture content are just a few of the factors that affect how the image segmentation method performs. Our invention introduces a rapid and efficient technique for eliminating blood vessels and hard exudates from a colour fundus picture of the eye. The simulation results on the retinal dataset demonstrate how the recommended technique may be used to retinal images and improve blood vessel and hard exudates detection in order to reduce human error or provide service in remote locations. Our project effort presents a substantial screening technique for the early detection of diabetic retinopathy. The proposed method computes more quickly to automatically detect the important clinical features of retinal images, including blood vessels, hard exudates, and optic disc. As a result, when the system receives a retinal picture, its output reveals the presence of exudates. A fundus camera may be used by an ophthalmologist to take retinal photographs of multiple patients while he is at an eye camp. Once the images are put into a system, any issues can be swiftly identified. This expedites the analytical process and boosts output.

12. FUTURE SCOPE

Maculopathy and Microaneurysm detection may be anticipated, and performance can be compared. Additionally, as part of our ongoing study, we wish to expand the scope of the recommended technique's applicability to cover more types of medical imaging, such OCTA pictures. In addition to the characteristics mentioned in the recommended approach, other factors can be considered.

13. APPENDIX

SOURCE CODE:

```
from flask import Flask, render_template, request, redirect, url_for
import numpy as np
import os
os.environ["CUDA_VISIBLE_DEVICES"] = "-1"
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
from cloudant.client import Cloudant

# basepath = os.path.dirname(__file__)
# print(basepath)
print(os.getcwd())
model = load_model(r'model/Updated-Xception-diabetic-retinopathy.h5')
client =
Cloudant.iam("ee288fae-8bbe-416f-b9a5-6b0c59bd45de-bluemix",

"MOuW96ZLdrjw4lHFV15pX7ZmIrSgs4VJqjS95Wp5D1VS",
connect=True)
myDB = client.create_database('retinopathy')
app = Flask(__name__)

# pages


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

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


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


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


@app.route('/registerUser', methods=['post'])
def registerUser():
    x = [x for x in request.form.values()]
    print(x)
    data = {
        '_id': x[1],
        'name': x[0],
        'pass': x[2]
    }
    print(data)
    query = {'_id': {'$eq': data['_id']}}

    docs = myDB.get_query_result(query)
    print(docs)

    if (len(docs.all()) == 0):
        url = myDB.create_document(data)
```

```
        return render_template('register.html', pred='Registration successful,
please login with your details')
    else:
        return render_template('register.html', pred="you are already
registered. please login with your credential")


@app.route('/loginUser', methods=['POST'])  # type: ignore
def loginUser():
    print(request.form)
    user = request.form['email']
    passw = request.form['password']
    print(user, passw)

    query = {'_id': {'$eq': user}}
    docs = myDB.get_query_result(query)
    print(docs)

    if (len(docs.all()) == 0):
        return render_template('login.html', pred="username not found")
    else:
        if (user == docs[0][0]['_id'] and passw == docs[0][0]['pass']):
            return redirect(url_for('prediction'))
        else:
            print('Invalid user')


@app.route('/predictImage', methods=['POST'])
def predictImage() -> object:
    # print(request.files)
    f = request.files['image']
    basepath = os.path.dirname(__file__)
    print(basepath)
    filepath = os.path.join(basepath, 'uploads', f.filename)  # type: ignore
    print(filepath)
```

```
    f.save(filepath)

    img = image.load_img(filepath, target_size=(299, 299))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)

    img_data = preprocess_input(x)
    print('finish1')
    _prediction = np.argmax(model.predict(img_data), axis=1)
    print('finish2')
    index = ['No Diabetic Retinopathy', 'Mild DR',
            'Moderate DR', 'Severe DR', 'Proliferative DR']

    result = str(index[_prediction[0]])
    print(result)
    return render_template('prediction.html', prediction=result)


# main driver function
if __name__ == '__main__':
    app.run(debug=False)
```

home.html

```
<!DOCTYPE html>
<html>

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

    <script>
```

```
document.addEventListener("DOMContentLoaded", function (event)
{
    const resultDiv = document.getElementById('resultDiv')
    const getInfoBtn = document.getElementById('getInfoBtn')
    const getInform = document.forms[0]
    getInfoBtn.addEventListener('click', function (e) {
        e.preventDefault()
        const options = {
            method: 'GET',
            headers: {
                'X-RapidAPI-Key':
'84a69684femsh1e293486e06ba0cp1825d9jsn6aa7c5b37425',
                'X-RapidAPI-Host': 'dietagram.p.rapidapi.com'
            }
        };

fetch(`https://dietagram.p.rapidapi.com/apiFood.php?name=${getInform["
food"].value}`, options)
            .then(response => response.json())
            .then(response => {

                if (response.dishes.length > 0) {
                    resultDiv.innerHTML = `<table class=table>
                    <tr>
                        <th>Name</th>
                        <th>Calories</th>
                        <th>Protien</th>
                        <th>Fat</th>
                        <th>carbon</th>
                    </tr>`+
                        response.dishes.map((el) =>
                            `<tr>
                                <td>${el.name}</td>
                                <td>${el.caloric}</td>
                                <td>${el.protein}</td>
```

```
                        <td>${el.fat}</td>
                        <td>${el.carbon}</td>
                    </tr>`
                ).join("") +
                `</table>`
            }
            else {
                window.alert("No Information found")
            }


        })
        .catch(err => window.alert("Maximum api calls reached"));
    })

    const getMedicinalDetails =
document.getElementById('getMedicinalDetails')
    getMedicinalDetails.addEventListener('click', function (e) {
        e.preventDefault()
        const form = document.forms[1]
        const options = {
            method: 'GET',
            headers: {
                'X-RapidAPI-Key':
'84a69684femsh1e293486e06ba0cp1825d9jsn6aa7c5b37425',
                'X-RapidAPI-Host':
'drug-info-and-price-history.p.rapidapi.com'
            }
        };


fetch(`https://drug-info-and-price-history.p.rapidapi.com/1/druginfo?drug=
${form["medicine"].value}`, options)
            .then(response => response.json())
            .then(response => {
```

```
                if (Array.isArray(response)) {
                    resultDiv.innerHTML = `<table class=table>
                    `+
                        Object.entries(response[0]).map(([key, value]) =>
                            (key !== 'packaging' && key !== 'openfda') ?
                                `<tr>
                                <th>${key}</th>
                                <td>${key === "active_ingredients" ?
value.map((el) => el.name).join(', ') : value}</td>
                                </tr>`: ''
                        ).join("") +
                        `</table>`
                }
                else {
                    window.alert(response)
                }
            })
            .catch(err => window.alert(err));
        })
    });

  </script>
</head>

<body>
  <div class="dflex f-row nav">
    <h3 class="navTitle">Diabetic Retinopathy Classification</h3>
    <div class="pages">
      <p class="page active">Home</p>
      <p class="page"><a href="/login"
class="no-decor">Login</a></p>
      <p class="page"><a href="/register"
class="no-decor">Register</a></p>
    </div>
  </div>
```

```
   <div class="dflex f-col">
      <img src="{{ url_for('static',filename='images/retinopathy.webp') }}"
alt="retinopathy image" class="image">
      <div class="dflex f-col desc">
         <div class="title">
            <p class="titleText">ABOUT PROJECT</p>
            <span class="underline"></span>
         </div>
         <div class="dflex f-row">
            <div class="dflex f-col descDiv">
               <h3>Problem</h3>
               <p class="text">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.
               </p>
            </div>
            <div class="dflex f-col descDiv">
               <h3>Solution</h3>
               <p class="text">In this project, we eill be building a Transfer
learning model that can detect and
                  classify types of diabetic retinopathy. A web app is
integrated with the model from where the
                  user can upload a Diabetic Retinopathy image like Mild
DR, severe DR, etc and see teh analysed
```

```
                results on User Interface
            </p>
        </div>
    </div>
</div>


</body>

</html>
```

login.html

```
<!DOCTYPE html>
<html>

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

<body>
    <div class="formContainer">
        <img src="{{ url_for('static',filename='svg/Farmer-512.webp') }}"
alt="profile" class="profile">
        <form action="{{ url_for('loginUser') }}" method="POST"
class="w-90">
            <input type="email" class="input" placeholder="Enter Email"
name="email" id="email"
                >
            <input type="password" class="input" placeholder="Enter
password" name="password" id="password"
                >
```

```
            <input type="submit" class="input button" value="Login">
        </form>
    </div>
    <p class="text" id="result">
        {% if pred %}
        {{ pred }}
        {% endif %}
    </p>
    <p class="text">Don't have an account? <a
href="/register">Register</a></p>
</body>

</html>
```

logout.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title></title>
  </head>
  <body style="text-align:center;">
      <h1>Successfully Logged out</h1><br />
      <a href="LoginPage.html">LOGIN</a>

  </body>
</html>
```

prediction.html

```
<!DOCTYPE html>
<html>

<head>
```

```
    <title>Home</title>
    <link rel="stylesheet" href="{{ url_for('static',filename='css/home.css')
}}">
</head>

<body>
    <div class="dflex f-row nav">
        <h3 class="navTitle">Diabetic Retinopathy Classification</h3>
        <div class="pages">
            <p class="page"><a href="/" class="no-decor">Logout</a></p>
        </div>
    </div>
    <div class="dflex f-col al-center">
        <div class="formContainer">
            <form action="{{ url_for('predictImage') }}" method="POST"
enctype="multipart/form-data">
                <img src="{{ url_for('static',filename='images/addImage.webp')
}}" alt="" class="addImage"
                    id="imagePreview">
                <input type="file" accept="image/*" name="image" id="image"
class="input">
                <input type="submit" value="Predict" class="input button">
            </form>
        </div>
        <p class="text" id="result">
            {% if prediction %}
            Result = {{ prediction }}
            {% endif %}
        </p>
        <img src="{{
url_for('static',filename='images/classesOfDisease.webp') }}" alt="types
of Retinopathy"
            class="retinopathyTypes">
    </div>
```

```
    <script>
        const resultContainer = document.getElementById('result')
        const selectedImg = document.getElementById('imagePreview')
        const imgInput = document.getElementById('image')

        imgInput.addEventListener('change', function (event) {
            selectedImg.src = imgInput.value
            if (event.target.files.length > 0) {
                let src = URL.createObjectURL(event.target.files[0]);
                selectedImg.src = src;
            }
        })
    </script>
</body>

</html>
```

register.html

```
<!DOCTYPE html>
<html>

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

<body>
    <div class="formContainer">
        <img src="{{ url_for('static',filename='svg/Farmer-512.webp ') }}"
alt="profile" class="profile">
        <form action="{{ url_for('registerUser') }}" method="POST"
class="w-90">
```

```
        <input type="text" class="input" placeholder="Enter Name"
name="name" id="name">
        <input type="email" class="input" placeholder="Enter Email"
name="email" id="email"
            >
        <input type="password" class="input" placeholder="Enter
password" name="password" id="password"
            >
        <input type="submit" class="input button" value="Register">
      </form>
   </div>
   <p class="text" id="result">
      {% if pred %}
      {{ pred }}
      {% endif %}
   </p>
   <p class="text">Already have an account? <a
href="/login">Login</a></p>
</body>

</html>
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-12013-1659366369

GITHUB DEMO LINK:

https://drive.google.com/file/d/1uOKoMI-StHwDKBXjZyGy-AbXqndh8WYh/view?usp=share_link