PROJECT DOCUMENTATION

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team Id: PNT2022TMID02274

Submitted By:

Sharveshwaran R - 2116190701200 - Team Lead

Shwetha M - 2116190701204

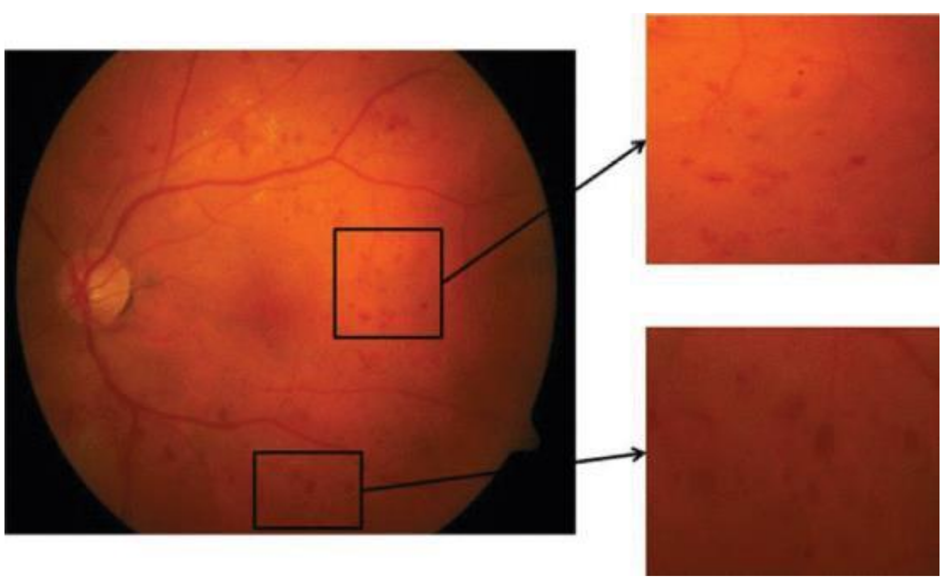
Sanjeev Krishnan R - 211619070186

Sneha K S - 2116190701206

Table of Contents

1. INTRODUCTION

The main causing of visual loss in the world is diabetic retinopathy. In the initial stages of this disease, the retinal microvasculature is affected by several abnormalities in the eye fundus such as the microaneurysms and/or dot hemorrhages, vascular hyper permeability signs, exudates, and capillary closures. Micro-aneurysm dynamics primarily increase the risk that the laser photo coagulation requires progression to the level. Diabetic retinopathy lesions are commonly accepted to be reversed and the progression of the retinopathy can only be slower during the early stages of the disease. The identification by repeated examination of patients affected of these initial lesions (mainly Micro aneurysms and small blood cells) is expected as a new possibility of improving retinopathy treatment. Floating and flashes, blurred vision, and loss of sudden vision can be common symptoms of diabetic retinopathy.



1.1 Project Overview

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

Transfer learning has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. We used Transfer Learning techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective.

1.2 Purpose

The Proposed work intends to automate the detection and classification of diabetic retinopathy from retinal fundus image which is very important in ophthalmology. Most of the existing methods use handcrafted features and those are fed to the classifier for detection and classification purpose. Recently convolutional neural network (CNN) is used for this classification problem but the architecture of CNN is manually designed. In this work, a genetic algorithm based technique is proposed to automatically determine the parameters of CNN and then the network is used for classification of diabetic retinopathy. The proposed CNN model consists of a series of convolution and pooling layer used for feature extraction. Finally support vector machine (SVM) is used for classification. Hyper-parameters like number of convolution and pooling layer, number of kernel and kernel size of convolution layer are determined by using the genetic algorithm. The proposed methodology is tested on publicly available Messidor dataset. The proposed method has achieved accuracy of 0.9867 and AUC of 0.9933. Experimental result shows that proposed auto-tuned CNN performs significantly better than the existing methods. Use of CNN takes away the burden of designing the image features and on the other hand genetic algorithm based methodology automates the design of CNN hyper-parameters.

2. LITERATURE SURVEY

2.1. EXISITING PROBLEM

Diabetic Retinopathy (DR) is a degenerative disease that impacts the eyes and is a consequence of Diabetes mellitus, where high blood glucose levels induce lesions on the eye retina. Diabetic Retinopathy is regarded as the leading cause of blindness for diabetic patients, especially the working-age population in developing nations. Treatment involves sustaining the patient's current grade of vision since the disease is irreversible. Early detection of Diabetic Retinopathy is crucial in order to sustain the patient's vision effectively. The main issue involved with DR detection is that the manual diagnosis process is very time, money, and effort consuming and involves an ophthalmologist's examination of eye retinal fundus images. The latter also proves to be more difficult, particularly in the early stages of the disease when disease features are less prominent in the images. Machine learning-based medical image analysis has proven competency in assessing retinal fundus images, and the utilization of deep learning algorithms has aided the early diagnosis of Diabetic Retinopathy (DR). This paper reviews and analyzes state-of-the-art deep learning methods in supervised, self-supervised, and Vision Transformer setups, proposing retinal fundus image classification and detection. For instance, referable, non referable, and proliferative classifications of Diabetic Retinopathy are reviewed and summarized. Moreover, the paper discusses the available retinal fundus datasets for Diabetic Retinopathy that are used for tasks such as detection, classification, and segmentation. The paper also assesses research gaps in the area of DR detection/classification and addresses various challenges that need further study and investigation.

2.2. REFERENCES

Fulong Ren 1 2 , Peng Cao 1 2 , Dazhe Zhao 1 2 , Chao Wan 3 macula localization, exudate candidate identification with vector quantization and exudate candidate classification with semisupervised learning. The proposed method and the state-of-the-art approaches are compared in terms of performance, and experimental results show the proposed system overcomes the challenge of the DME grading and demonstrate a promising effectiveness. Kangrok Oh, Hae Min Kang, Dawoon Leem, Hyungyu Lee, Kyoung Yul Seo & Sangchul Yoon.They measure image-wise RSD values using the test model outputs from the ten runs of cross-validation tests. Consequently, average RSD values for both DR detection systems based on ETDRS 7SF and F1–F2 images are reported Silva et al. demonstrated that peripheral lesions identified on UWF imaging are associated with the increased risk of DR progression37. Those pioneering studies33,34,35,36,37 regarding the UWF imaging for DR severity evaluation utilized capturing devices from Optos. The wide-field scanning laser ophthalmoscopy (SLO) by Optos provides a single image covering nearly 200∘ of the retina18. During transforming the wide-field image of the spherical eye into the 2-D image, small lesions may be inconspicuous due to distortion18. Furthermore, eyelashes and eyelids cover the superior and inferior periphery of the retina in some cases32. Aiello et al.33 demonstrated that the ETDRS 7SF photography and corresponding fields in the UWF photography have moderate to substantial agreements for DR severity evaluation. Poornima S V, Parvatha Lakshmi B, Nishchala T K, Umamakeswari A automated the detection of diabetic retinopathy, thereby eliminating errors culminated by human measurement. Fundus images obtained from HRF database [2] have been used for this study. Early Detection of Diabetic Retinopathy by Using Deep Learning Neural Network Mohamad Hazim Johari1, Hasliza Abu Hassan2, Ahmad Ihsan Mohd Yassin1*, Nooritawati Md Tahir1, Azlee Zabidi1, Zairi Ismael Rizman3, Rahimi Baharom1, Norfishah Abdul Wahab1 he data set used were retrieved from MESSIDOR database and it contains 1200 pieces of fundus images. The images were filtered based on the project needed. There were 580 pieces of images types .tif has been used after filtered and those pictures were divided into 2, which is Exudates images and Normal images. On the training and testing session, the 580 mixed of exudates and normal fundus images were divided into 2 sets which is train-ing set and testing set. The result of the training and testing set were merged into a confusion matrix. The result for this project shows that the accuracy of the CNN for training and testing set was 99.3% and 88.3% respectively Deep Learning Fundus Image Analysis for Diabetic Retinopathy and Macular Edema Grading Jaakko Sahlsten1, Joel Jaskari1, Jyri Kivinen1, Lauri Turunen2, Esa Jaanio2, Kustaa Hietala3 & Kimmo Kaski1,*they first present the details of smartphone-based portable retinalimaging systems available on the market to compare their features and image qualities.Second, they introduce the Field of View (FoV) determination process of each smartphone-based retinal imaging system using a circular test pattern. Third, they introduce the layout of the adopted deep learning architecture for DR detection Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms LIFENG QIAO 1, YING ZHU 2, AND HUI ZHOU 2.To propose the Prognosis of Microaneurysm and early diagnosis system for non - proliferative diabetic retinopathy (PMNPDR) utilizing a deep convolutional neural network for semantic segmentation of fundus images which can increase the efficiency and accuracy of NPDR. • Maximum matching filter response (MFR) mutual infor-mation (MI) and maximum Gaussian answer laplacian (LoG) in the 2-dimension function space utilizing Differential Evolution which, has not been previously explored in the detection of lesions. • The experimental results have been performed based on the datasets (https://ieee-dataport.org) [25].

2.3. PROBLEM STATEMENT DEFINITION

Diabetic Retinopathy (DR) is common complication of diabetes mellitus, which will cause lesions on the retina that affects vision. If it is not detected early, it can lead to blindness. Unfortunately, DR is not a reversible proves, and the given treatment will only give us a sustain vision. DR early detection and treatment can significantly reduce the risk of vision loss.

WHAT? In contrast to computer-aided diagnosis systems, the manual / human-based diagnosis process of DR retina fundus images by doctors (ophthalmologists) is time-consuming, labor-intensive, expensive, and prone to error.

WHY? Diabetes-related retinopathy is brought on by high blood sugar levels harming the eye's iris. which could result in a permanent loss of vision.

WHEN? Early on, the DR has no symptoms, but later on, the vessels may start to leak a tiny amount of blood into your retina.

WHERE? Blurred vision, Distorted vision will occur.

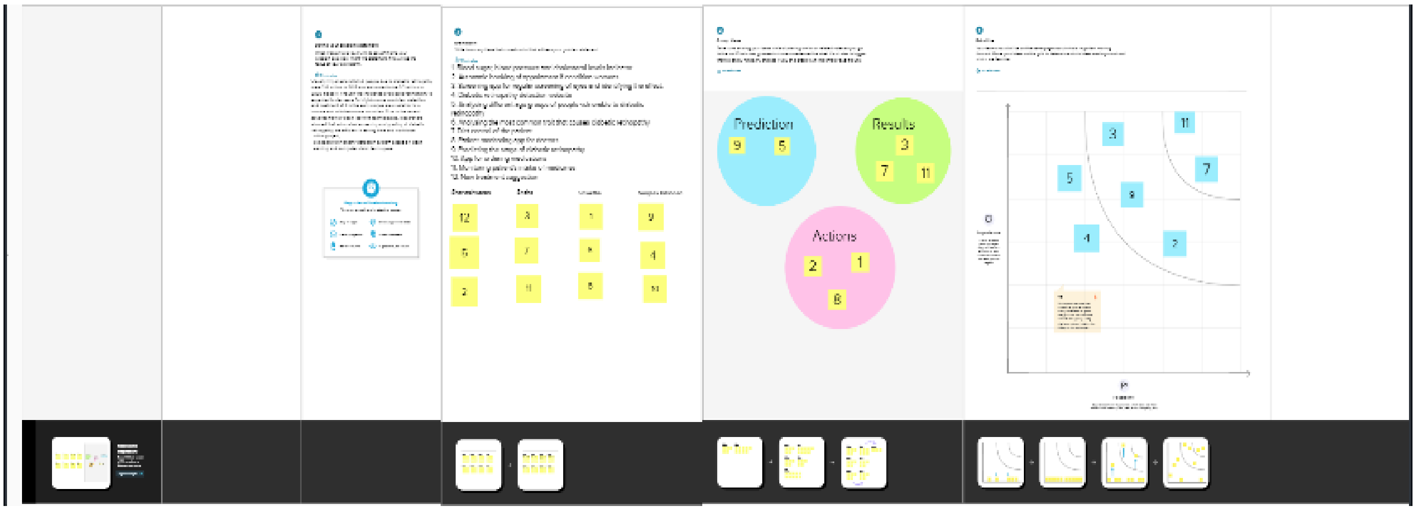
WHO? It is common among the Diabetic patients.

HOW? The manual early detection of this DR is a challenging tas

3.IDEATION PHASE & PROPOSED SOLUTION

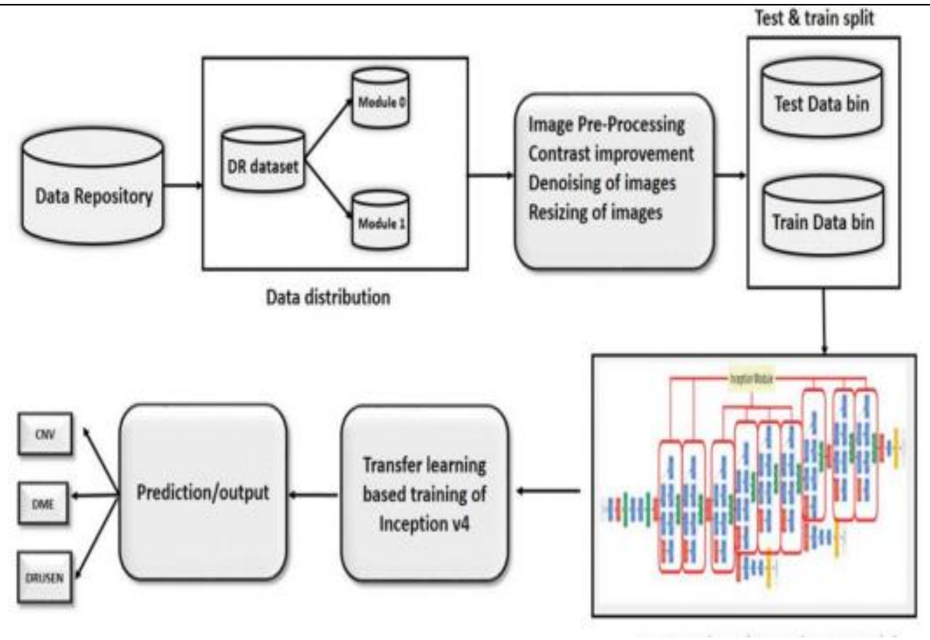
3.1 Empathy Map Canvas

3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Diabetes is a globally prevalent disease that can cause visible microvascular complications such as diabetic retinopathy in the human eye retina, the images of which are today used for manual disease screening and diagnosis. This labour-intensive task could greatly benefit from automatic detection using deep learning technique
2.	Idea / Solution description	Here we present a deep learning system that identifies referable diabetic retinopathy comparably or better than presented in the previous studies, although we use only a small fraction of images (less than 1/4th) in training but are aided with higher image resolutions.
3.	Novelty / Uniqueness	We are providing novel results for five different screening and clinical grading systems for diabetic retinopathy including state of the art results for more accurately classifying images according to clinical five grade diabetic retinopathy

4.	Social Impact / Customer Satisfaction	This deep learning model can be used to identify people with diabetic retinopathy and Diagnose the clinical grade of diabetic retinopathy in them
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	This deep learning system could increase the cost effectiveness of screening and diagnosis attaining higher than recommended performance and that the system could be applied in clinical examinations requiring finer grading

3.4 PROPOSED SOLUTION FIT

Project Title: Deep Learning Fundus Image Analysis For early detection of Diabetic Retinopathy.

Project Design Phase-I = Problem-Solution Fit Template

Project ID : IBM-Project-18407-1659684768

ne CS, fit into CC

1. CUSTOMER

For diabetics, early detection is crucial because diabetic retinopathy is permanent. The patient's fundus image can be used to identify diabetic retinopathy and be kept in the database. This serves a greater purpose than a manual examination.

6. CUSTOMER

Because diabetic retinopathy does not have any obvious symptoms, people are unaware they have the condition. Many people are unaware of diabetic retinopathy and its harmful effects.

5. AVAILABLE

Proliferative diabetic retinopathy can be treated with laser therapy, and some forms of maculopathy can be stabilized with laser therapy as well. eye injections to cure your sight-threatening severe maculopathy.

plore AS, differentiate

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

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

The issue is that it is impossible to treat severe diabetic retinopathy. Furthermore, the severity of diabetic retinopathy causes serious eye conditions that might lead to blindness. Therefore, if the patient has diabetes, early identification is crucial.

9. PROBLEM ROOT CAUSE RC

The retina, a layer of light-sensitive tissue at the rear of the inner eye, experiences alterations in its blood vessels as a result of diabetes. The blood vessels in the retina of some patients with diabetic retinopathy may enlarge and leak fluid. Others experience the aberrant growth of new blood vessels on the retinal surface.

7. BEHAVIOUR BE

Using the pictures from the fundus, this model aids in the early diagnosis of diabetic retinopathy. I
The manual examination takes longer than this.
Additionally, accuracy is higher compared to other methods.

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

Identify strong TR & E

3. TRIGGERS TR

Patients with diabetic retinopathy experience these triggers:

You may notice spots or black strings in your vision (floaters) distorted vision.

unstable eyesight.
Visionary voids or patches of darkness.
loss of vision

4. EMOTIONS: BEFORE / AFTER EM

Before: Fear and anxiety are examples of negative emotional reactions. concerns with self-perception, guilt, rage, insecurity, and vulnerability.
After: Patients feel more hopeful as a result of early discovery and diagnosis.

10. YOUR SOLUTION SL

Our approach uses a deep learning model with fundus images to identify diabetic retinopathy severity in patients and to make an appropriate diagnosis following an early detection.

8.CHANNELS OF BEHAVIOUR CH

Patients with diabetes must have their eyes examined at regular intervals. Only then may retinopathy be identified early and a correct diagnosis made.

4.REQUIREMENT ANALYSIS

4.1. Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Information	Enter user details. The image of the user must be uploaded
FR-4	User input	Upload the user's retinal scan
FR-5	Result	Output is displayed in text area and option to download the report in .pdf format is available

4.2. Non-functional Requirements:

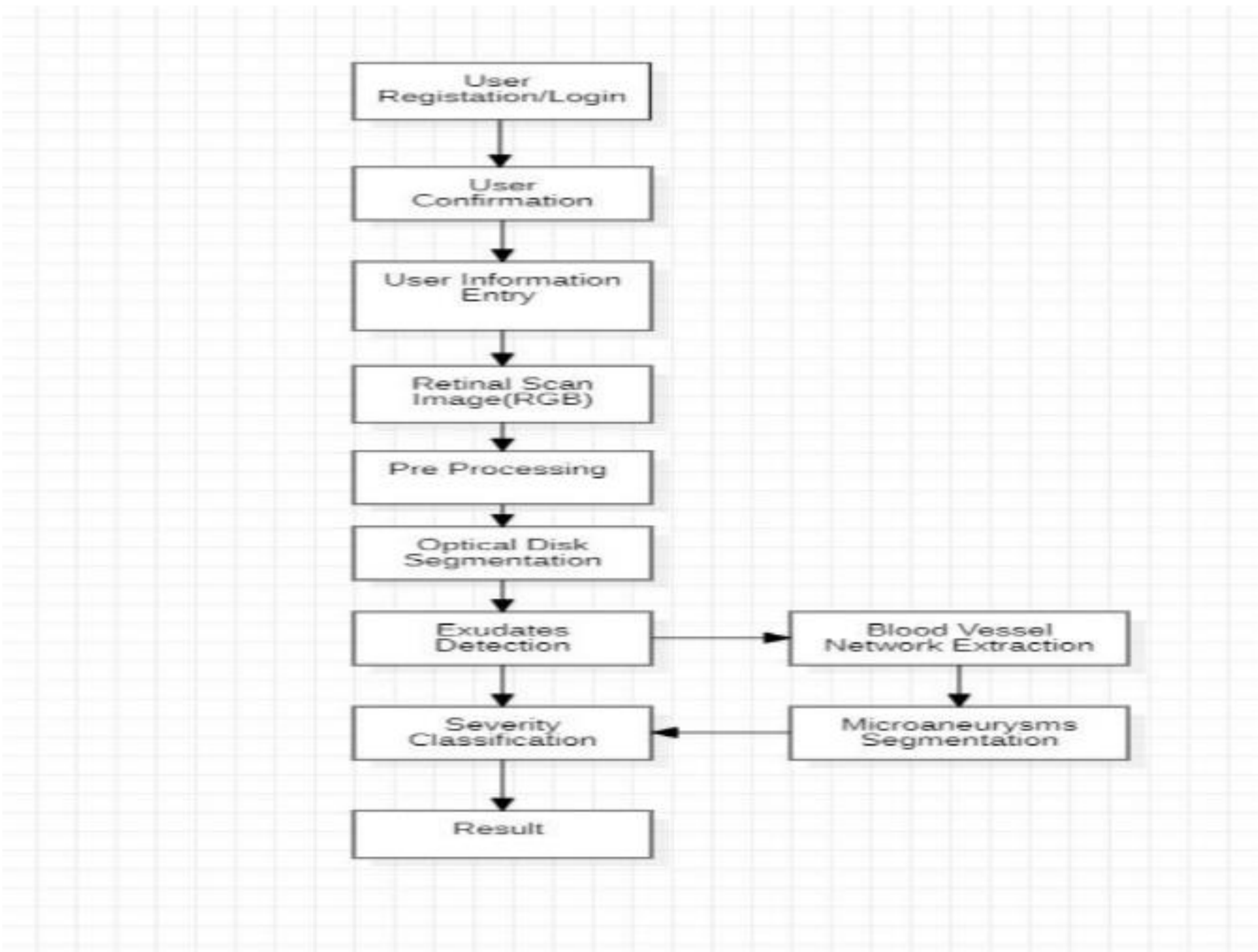
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This environment is user friendly,easily accessible and intuitive for people of
NFR-2	**Security**	The user's details and reports are stored and maintained securely for later retrieval
NFR-3	**Reliability**	Even if the system fails the downtime is super low so the system recovers quickly
NFR-4	**Performance**	The processing time of the web application (I.e time taken to enter the details and fetch results) is low

NFR-5	**Availability**	It is available around the clock 24x7
NFR-6	**Scalability**	The app should be capable of being expanded to a larger user base and still function the way it is intended to be

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 TECHNOLOGY ARCHITECTURE

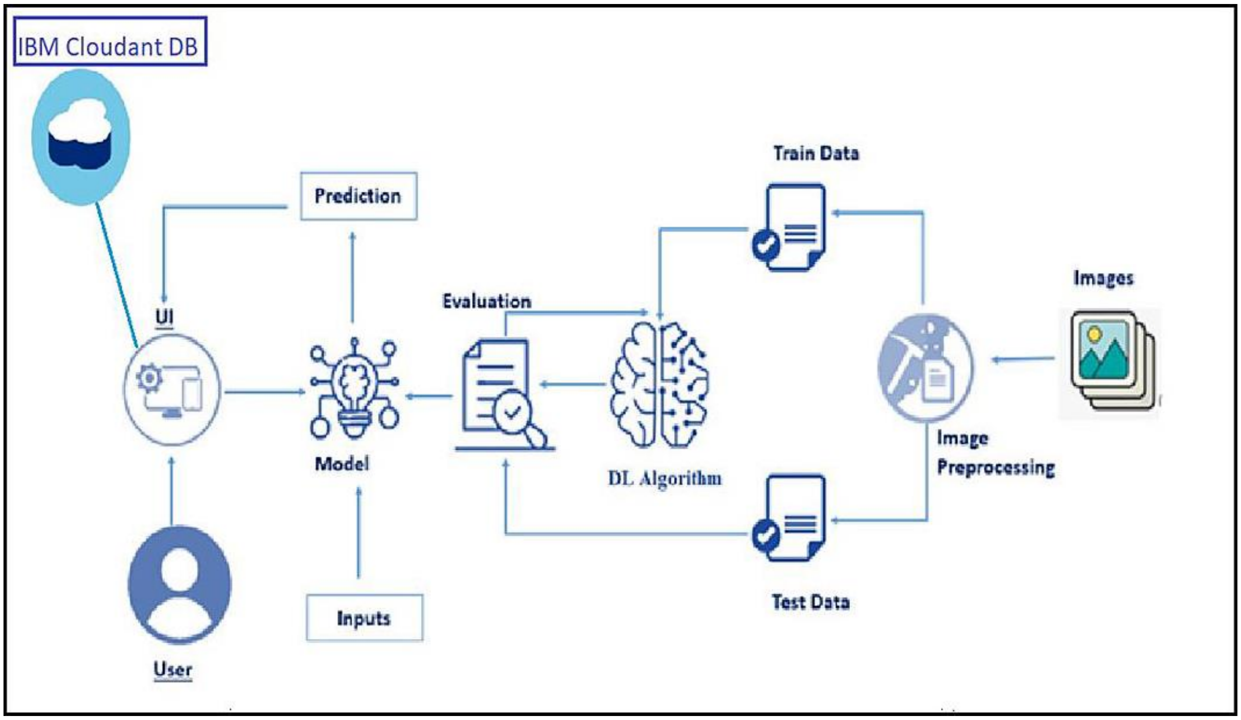


Table-1: Components& Technologies

1.	User Interface	Web UI	HTML, CSS, JavaScript, Python
2.	Application logic-1	Image Preprocessing	Keras,Tensorflow,Numpy
3.	Application logic-2	CNN Model	Keras,Tensorflow,Numpy

4.	Application logic-3	Web UI Application	Flask
5.	Database	DR Images (Jpeg,Png,Jpg,Etc.,)	Uploads Folder
6.	File storage	File Storage Requirements (Only If Necessary)	IBM Block Storage, GoogleDrive
7.	External Api	Keras	Image Processing API
8.	Deep Learning Model	Inception V3 Architecture	Pre-Trained Convolution NeuralNetwork Model
9.	Infrastructure (Server)	Application Deployment on Webserver	Flask-A PythonWSGI HTTP Server.

Table-2: Application characteristics

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Flask	Flask Frameworks
2.	Security Implementations	CSRF Protection,Secure Flag For Cookies	Flask-WTF, Session Cookie Secure
3.	Scalable Architecture	Micro-Services	Micro Web Application FrameworkBy Flask

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Patient (Webuser)	Registration	USN-1	I can register as a user on the website with eitheran email address or a phone number and password.	I can createmy account.	High	Sprint-3
	Login	USN-2	With theprovided Login credentials, I canaccessthe website as a user.	I can log in andaccess myaccount .	High	Sprint-3
	Upload image	USN-3	I can post my data as a userin formats likepdf and doc.	I can uploadmy data.	Medium	Sprint-3
Administration (Web developer)	Admin Login	USN-4	I can log in to the website as theadmin and analyze the user information .	I can log in and analyze the user data.	High	Sprint-3
	Data collection	USN-5	I can gatherthe dataset forthe DR fromthe source as anadmin.	I can collect the dataset.	Low	Sprint-1
	Create model	USN-6	I can buildthe model andtrain it using the dataset as an administrator to makepredictions.	I can create andtrain the model.	High	Sprint-1
	Test the model	USN-7	I canevaluate the model's predictive abilities as an admin.	I can testthe model.	High	Sprint-2
Patient (Web user)	Diagnosis	USN-8	I can access the application's diagnosisresults as a userand continue with treatments..	He/she can get the resultsand continue the treatment.	High	Sprint-2

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Working with Dataset	USN-1	To download and import the dataset along withnecessary libraries	5	High	Sharveshwaran R
		USN-2	To analyze the data and handle missing data	5	High	Sanjeev Krishnan R
		USN-3	To perform data visualization and finddependent and independent features	5	Low	Sneha K S
		USN-4	To perform feature scaling and split dataset intotrain and test	5	Medium	Shwetha M
Sprint-2	Model Decision	USN-5	To train and test CNN model and find the accuracy	10	Medium	Sharveshwaran R Shwetha M
		USN-6	To predict the results using CNN model	10	Medium	Sanjeev Krishnan R Sneha K S
Sprint-3	Building web application	USN-7	Creating a web page for content display	10	High	Shwetha M Sneha K S
		USN-8	Creating python script for prediction andrendering to web page	10	High	Sharveshwaran R Sanjeev Krishnan R
Sprint-4	Export Application	USN-9	Run and export the application	20	High	Sharveshwaran R Shwetha M Sanjeev Krishnan R Sneha K S

Sprint	Total story point	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)

Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

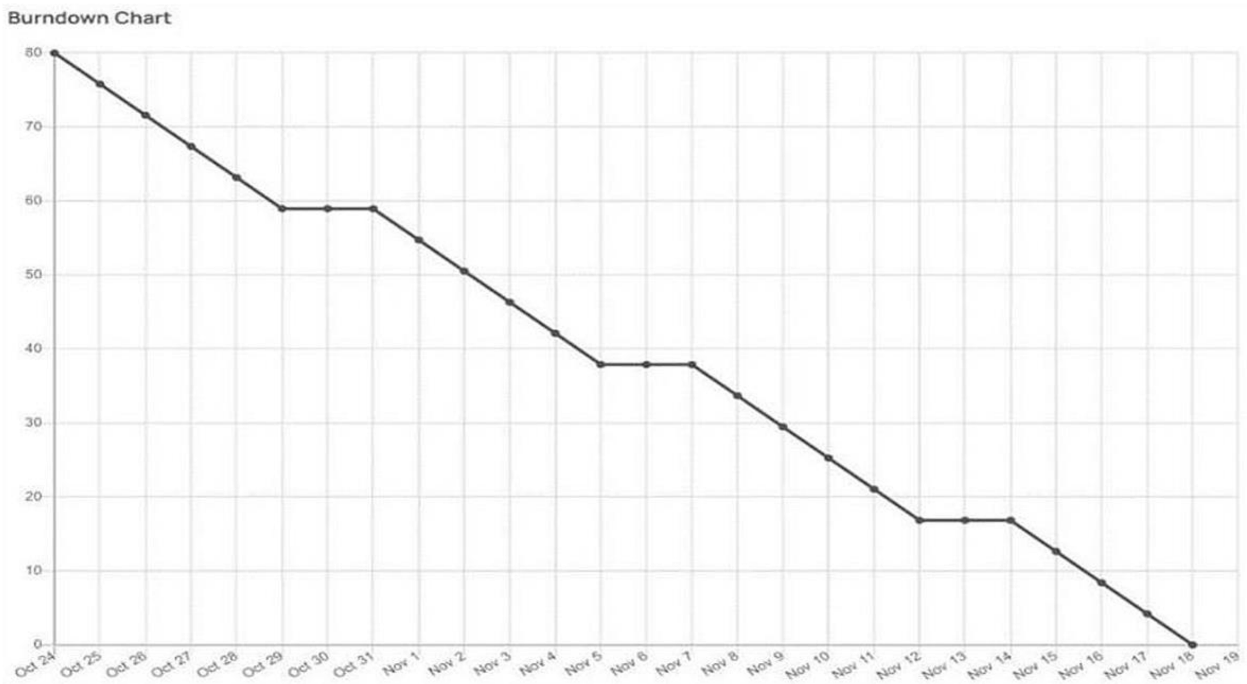
Velocity:

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

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

AV=20/6=3.33points per day.

6.2 Burn Down Chart & JIRA :



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

7.CODING AND SOLUTION:-

Feature 1:-

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

Feature 2:-

We have devloped a multilayer deep convolutional nueral network that classifies the user image of a eye to which extense has the disease diabetics has been affected.The model will classify the images into 5 categories of diabetics and report them on asking for prediction. We have also devloped a messaging service for recieiving message for the type of diabetics.

8.TESTING:-

8.1 TEST CASES:-

8.2 USER ACCEPTANCE TESTING:-

1. Purpose of Document:-

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

2. Defect Analysis:-

This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity4	Subtotal
By Design	5	4	2	3	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	15	30

Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	5	2	1	8
Totals	17	14	13	21	65

3. Test-CaseAnalysis

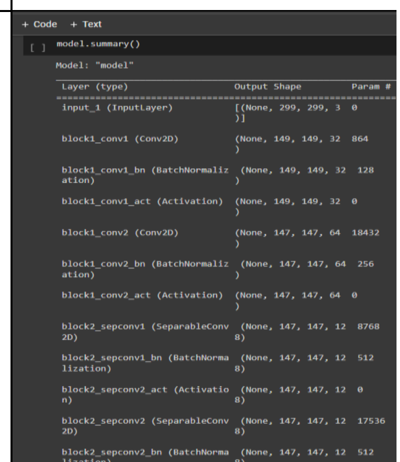
This report shows the number of test cases that have passed, failed,and untested.

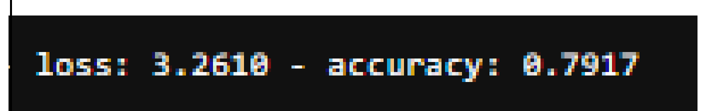
Section	TotalCases	Not Tested	Fail	Pass
PrintEngine	9	0	0	9
ClientApplication	45	0	0	45
Security	2	0	0	2
Out-sourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4
VersionControl	2	0	0	2

9.RESULTS:-

9.1 Performance Metrics:-

Model Performance Testing:

<table>
<tr><th>S. NO</th><th>Parameter</th><th>Values</th><th>Screenshot</th></tr>
<tr><td>1.</td><td>Model Summary</td><td>Total params: 21,885,485
Trainable params: 1,024,005
Non-trainable params: 20,861,480</td><td>
</td></tr>
<tr><td>2.</td><td>Accuracy</td><td>Training Accuracy – 0.7917

Validation Accuracy – loss 3.2610</td><td>
</td></tr>
<tr><td>3.</td><td>Confidence Score(Only Yolo Projects)</td><td>Class

Detected -

Confidence

Score -</td><td>--
--
--
--</td></tr>
</table>

Project team shall fill the following information in model performance testing template.

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

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

First, deep learning is well-suited for image analysis tasks. This is because deep learning algorithms can automatically learn features from images, which is essential for accurate image analysis.

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

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

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

Finally, deep learning is robust. This means that it is less likely to overfit to the data, which is important for medical image analysis where data is often limited.

10.2 DISADVANTAGES

There are several disadvantages of deep learning for early detection of diabetic retinopathy. One disadvantage is that deep learning requires a large amount of data to train the models. This can be a challenge for researchers who do not have access to a large dataset. Another challenge is that deep learning models can be very complex, which can make them difficult to interpret. Finally, deep learning models can be computationally intensive, which can make them difficult to deploy in resource-limited settings.

11. CONCLUSION

Diabetic retinopathy (DR) is a leading cause of blindness in the United States. Early detection and treatment of DR is critical to preventing vision loss. However, DR is often asymptomatic in its early stages, making it difficult to detect. Deep learning (DL) is a type of artificial intelligence that can be used to automatically detect patterns in data. DL has been shown to be effective for detecting DR in images of the retina. In this study, a DL algorithm was used to automatically detect DR in fundus images. The algorithm was able to accurately detect DR in early stages, before it is symptomatic. This could potentially lead to earlier diagnosis and treatment of DR, which could help to prevent vision loss.

12. FUTURE SCOPE

There is a great potential for deep learning in fundus image analysis for early detection of diabetic retinopathy. However, there are a few challenges that need to be addressed. First, the current data sets are small and lack diversity. Second, the images are often low quality and need to be pre-processed before they can be used for deep learning.

Third, the ground truth labels for the images are often not available. Finally, the current deep learning models are not able to generalize well to real-world data.

13. APPENDIX

app.py:-

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
from flask import Flask, request,flash, render_template, redirect,url_for
from cloudant.client import Cloudant
from twilio.rest import Client
model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
app = Flask(__name__)
app.secret_key="abc"
app.config['UPLOAD_FOLDER'] = "User_Images"
# Authenticate using an IAM API key

client = Cloudant.iam('c2122ee4-b4d5-4cd4-a15a-26af8dc9533e-bluemix',
                      '1HTTyEGgmFvrcaEzQqHflsuHpqZ3Y4UUsChgqXcIW_Ze',
                      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')
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 = True
    app.run()


<div
class="mb-3
d-flex
justify-
content-
center">
<a
href="login"
class="nav-
link">
Already
Registered:
Login
Here</a>
```

Python Notebook screenshots:-

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

In [ ]: mkdir ~/.kaggle

mkdir: cannot create directory '/root/.kaggle': File exists

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

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

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

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

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

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

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

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

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

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

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

In [ ]:
```
from glob import glob
```

In [ ]:
```
import numpy as np
```

In [ ]:
```
import matplotlib.pyplot as plt
```

In [ ]:
```
imageSize=[299,299]
```

In [ ]:
```
trainPath=r"/content/preprocessed dataset/preprocessed dataset/training"
```

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

In [ ]:
```
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

In [ ]:
```
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [ ]:
```
training_set=train_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/training',target_size=(299,299),batch_size=
```

```
Found 3662 images belonging to 5 classes.
```

In [ ]:
```
test_set=test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',target_size=(299,299),batch_size=32,clas
```

```
Found 3662 images belonging to 5 classes.
```

In [ ]:
```python
test_set=test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',target_size=(299,299),batch_size=32,cla
```

```
Found 734 images belonging to 5 classes.
```

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

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xception/xception_weights_tf_dim_ordering_tf_kernels_notop.h
5
83683744/83683744 [==============================] - 0s 0us/step
```

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

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

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

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

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

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


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


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

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

 block1_conv1_act (Activation)  (None, 149, 149, 32  0           ['block1_conv1_bn[0][0]']
                                )

 block1_conv2 (Conv2D)          (None, 147, 147, 64  18432       ['block1_conv1_act[0][0]']
                                )

 block1_conv2_bn (BatchNormaliz  (None, 147, 147, 64  256        ['block1_conv2[0][0]']
 ation)                         )

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

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

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

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

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

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

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

```
 alization)                      )

 block14_sepconv2_act (Activati  (None, 10, 10, 2048  0          ['block14_sepconv2_bn[0][0]']
 on)                             )

 flatten (Flatten)               (None, 204800)       0          ['block14_sepconv2_act[0][0]']

 dense (Dense)                   (None, 5)            1024005    ['flatten[0][0]']

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

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

```
In [ ]: r=model.fit_generator(training_set,validation_data=test_set,epochs=30,steps_per_epoch=len(training_set)//32,validation_steps=len(test_set)//32)
```

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

GITHUB LINK:- https://github.com/IBM-EPBL/IBM-Project-14061-1659539858

DEMO LINK:- https://github.com/IBM-EPBL/IBM-Project-14061-1659539858/tree/main/Final%20Deliverables/Demonstration%20Video