PROJECT REPORT

DEEP LEARNING FUNDUS IMAGE ANALYSIS FOR EARLY DETECTION OF DIABETIC RETINOPATHY

TEAM ID : PNT2022TMID09991

TEAM MEMBERS : KIRTHIK S(19104086) [TEAM LEADER]

HARI PRIYA C(19104066)

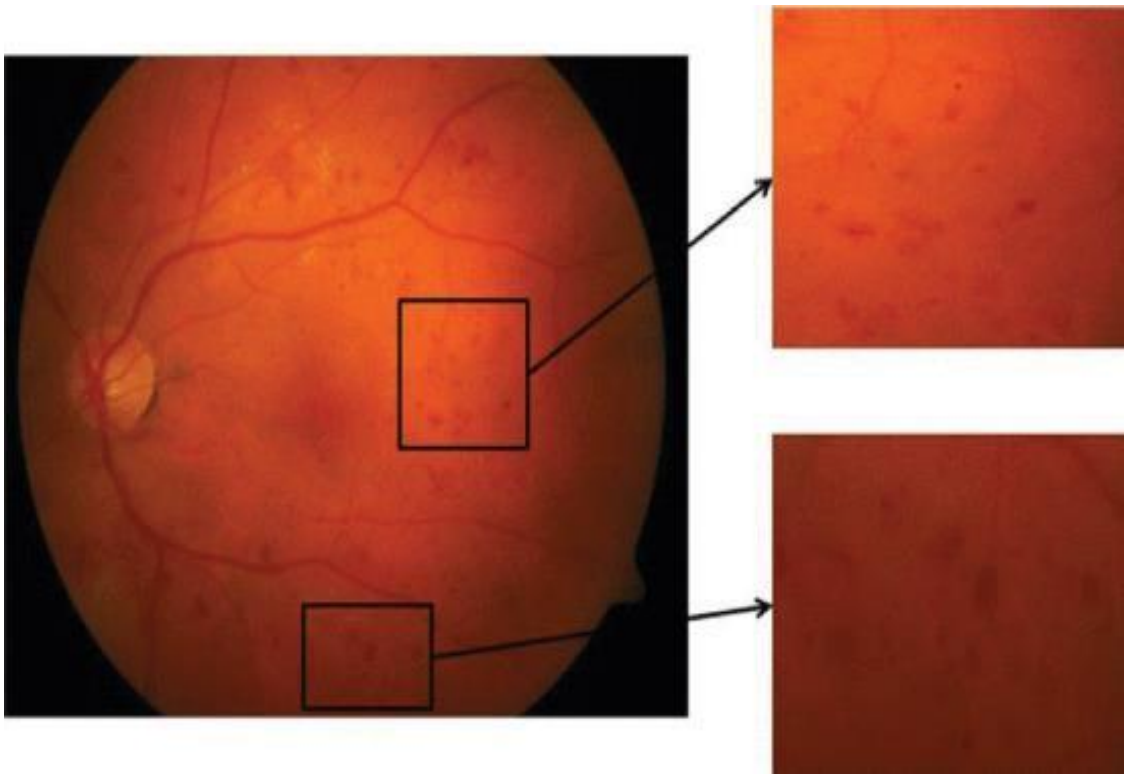
OVIYA R(19104121)

OYNDRISHA RAKSHIT(19104122)

1.INTRODUCTION:

Diabetic Retinopathy (DR) is a common complication of diabetic mellitus, which causes lesions on the retina that affect visions. 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 the vision loss. The manual diagnosis process of DR retina fundus images by ophthalmologists is the time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis system.

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 technique 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.



1.1 Project Overview:

Diabetic retinopathy is the most common microvascular complications in diabetes, for the screening of which the retinal imaging is the most widely used method due to its high sensitivity in detecting retinopathy. The evaluation of the severity and degree of retinopathy associated with a person having diabetes, is currently performed by medical experts based on the fundus or retinal image of the patients eyes. As the number of patients with diabetes is rapidly increasing, the number of retinal images produced by the screening programmes, it will also increase, which in turn introduces a large labor-intensive burden on the medical experts as well as cost to the health care services. This could be alleviated with an automated system either as support for medical experts' work or as full diagnosis tool. There are two recent studies have investigated the use of deep learning system in automated detection of diabetic retinopathy.

1.1 **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 method 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 serious of convolution and pooling layer used for feature extraction. Finally, support vector machine (SVM) is used for classification. Hyper parameters like number of convolutions 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 & AUC of 0.9933. Experimental result shows that proposed auto-tuned CNN performs significantly better than the existing method. 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:

EXISTING PROBLEM

Diabetic Retinopathy (DR) is a degenerative disease that impacts the eyes and is a consequence of diabetics 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 sustaning the patients 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 manual diagnosis progress is vey time, money, effort consuming and involves an ophthalmologist's examination of eye retinal fundus images. The late also proves to be more difficult, particularly in the early stages of the disease when the disease features are less prominent in accessing retinal fundus images and utilization of deep leaning algorithm has aided the early diagnosis of diabetic retinopathy (DR). This paper reviews and analysis state of the art deep learning methods in supervised, self-supervised and vision transformer setup proposing retinal fundus image classification and detection. For instances, referable, non referable and proliferative classifications of diabetic retinopathy are reviewed and summarized. More over the paper discusses the available retinal fundus data sets for diabetic retinopathy that are used for the tasks such as detection, classification and segmentation. The paper also access research gaps in the area of diabetic retinopathy detection or classification and addresses various challenges that need further study and investigation.

REFERENCES:-

Survey 1:

AUTHORS: Mohammad Z. Atwany , Abdulwahab H. Sahyoun , And Mohammad Yaqub (March 22).

TITLE: 'Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey.'

METHODS: This paper reviews and analyzes state-of- theart deep learning methods in supervised, self-supervised, and Vision Transformer setups, proposing retinal fundus image classification and detection. For instance, referable, nonreferable, 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.

Survey 2:

AUTHORS: Mohamad Hazim Johari , Hasliza Abu Hassan , Ahmad Ihsan Mohd Yassin (July 2018).

TITLE: 'Early Detection of Diabetic Retinopathy by Using Deep Learning Neural Network.'

METHODS: This project presents a method to detect diabetic retinopathy on the fundus images by using deep learning neural network. Convolution Neural Network (CNN) has been used in the project to ease the process of neural learning. The 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 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 training 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.

Survey 3:

AUTHOR: Recep Emre Hacisoftaoglu (Dec 2019).

TITLE: 'Deep Learning Frameworks For Diabetic Retinopathy Detection Using Smartphone-Based Retinal Imaging Systems.'

METHODS: In this thesis, we first investigate the smartphone-based portable ophthalmoscope systems available on the market and compare their Field of View and image quality to determine if they are suitable for Diabetic Retinopathy detection during a general health screening. Then, we propose automatic Diabetic Retinopathy detection algorithms for smartphone-based retinal images using deep learning frameworks, AlexNet and GoogLeNet. To test our proposed methods, we generate smartphone-based synthetic retina images by simulating the different Field of View with masking the original image around the optic disk and cropping it.

Survey 4:

AUTHORS: Lei Lu , Ying Jiang , Ravindran Jaganathan , and Yanli Hao. (Jan 2019).

TITLE: 'Current Advances in Pharmacotherapy and Technology for Diabetic Retinopathy: A Systematic Review.'

METHODS: Direct injections or intra virtual antiinflammatory and anti angiogenesis agents are widely used pharmacotherapy to effectively treat DR and diabetic macular edema (DME). However, their effectiveness is short term, and the delivery system is often associated with adverse effects, such as cataract and increased intraocular pressure.

Further, systemic agents and plants-based drugs have also provided promising treatment in the progression of DR. Recently, advancements in pluripotent stem cells technology enable restoration of retinal functionalities after transplantation of these cells into animals with retinal degeneration. This review paper summarizes the developments in the current and potential pharmacotherapy and therapeutic technology of DR. Literature search was done on online databases, PubMed, Google Scholar, clinitrials.gov, and browsing through individual ophthalmology journals and leading pharmaceutical company websites.

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 timeconsuming, 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.

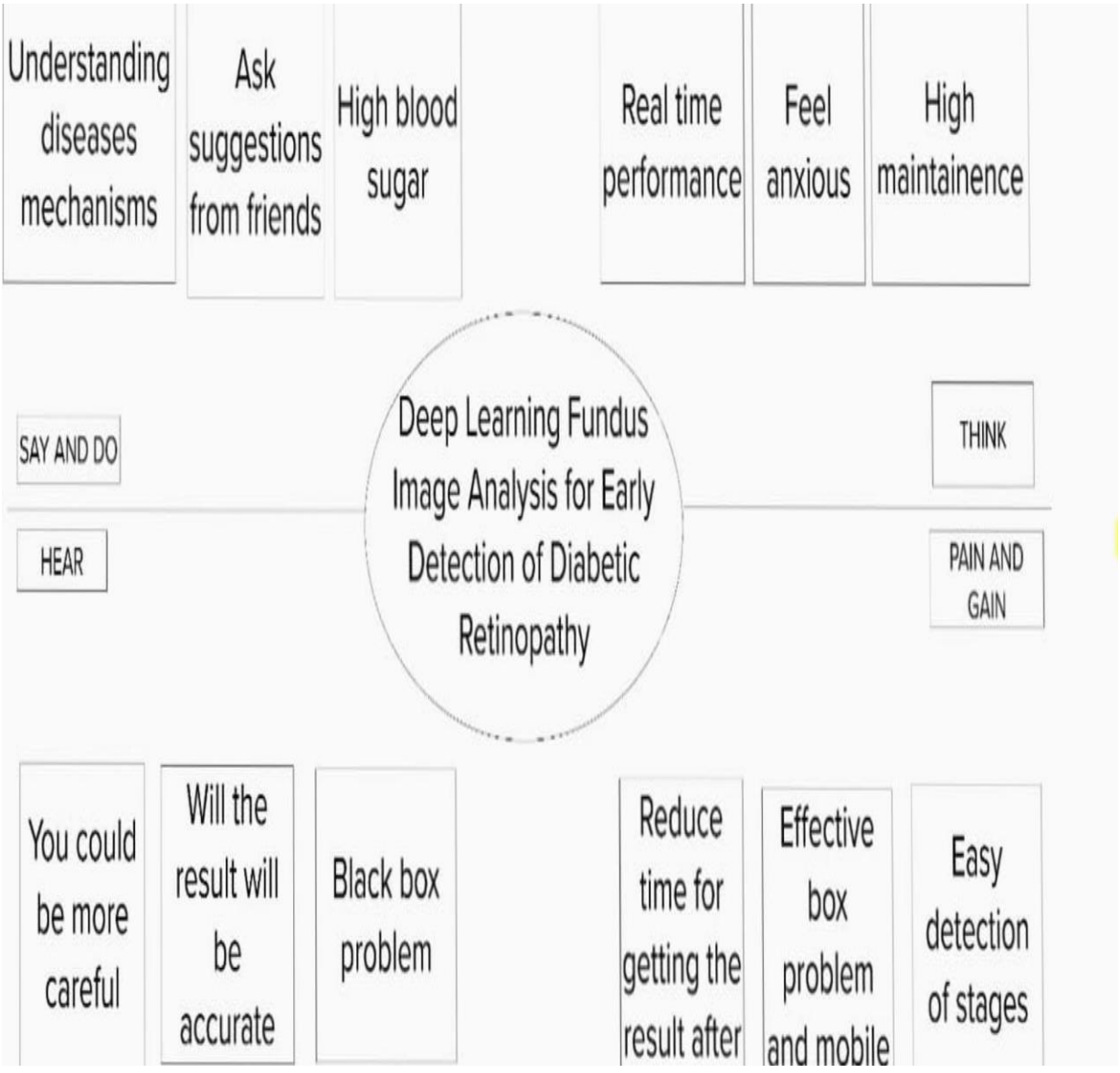
HOW ? The manual early detection of this DR is a challenging task.

OBJECTIVES :

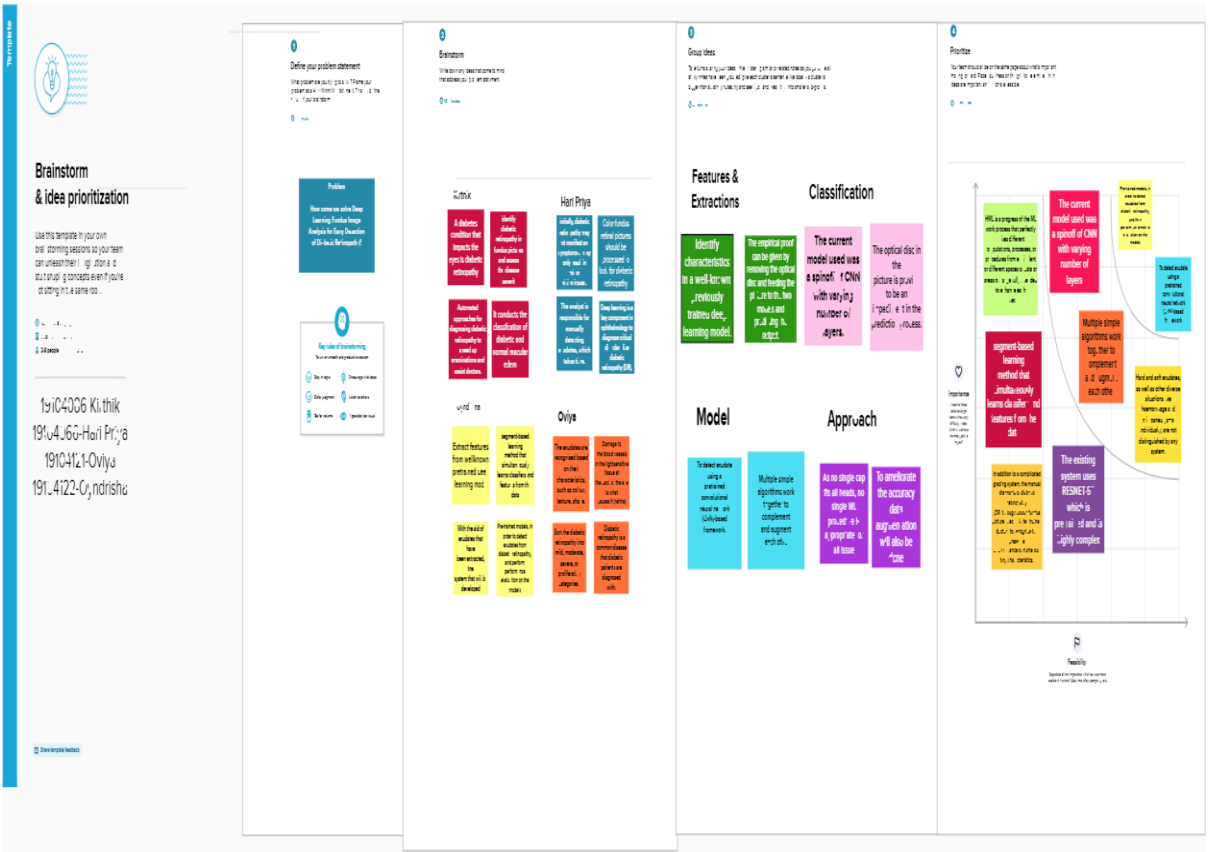
The primary goal is to identify diabetic retinopathy by processing retinal images. Transfer learning has arose as one of the most popular techniques that has enhanced performance in many areas, notably in the analysis and classification of medical images. We used transfer learning techniques that are more frequently used in medical image analysis and have been extremely effective, including such Inception V3, Resnet50, and Xception V3.

IDEATION AND PROPOSED SOLUTION:

EMPATHY MAP:



IDEATION AND BRAINSTORMING:



PROPOSED SOLUTION:

S.NO	Parameter	Description
1	Problem Statement (Problem to be solved)	Diabetic retinopathy is a state which is due to the damage of blood vessels of the retina. Considering the fact that Retina is the sensitive part it can result in blurry, less intense eye sight and it can also result in disappearing of eye sight. The diabetic retinopathy may cause no symptoms at in its earliest stages, They initial symptoms may be barely noticeable or mild. As time goes on, the state of this issue can worsen and lead to partial and then complete blindness to the individual which must be taken care of beforehand to get better at early stages. Thus, early detection of the diabetic retinopathy is highly recommendable.
2	Idea / Solution description	To avoid complication due to late identification of the disease, we develop a Deep learning system that can detect early-to-late stages of diabetic retinopathy by using Fundus image as dataset for training and testing the model. The deep learning models like Resent-50, Alexnet, VGG16, Google-Net, U-Net are under study as of now. After completion of data pre-processing, the model will be trained and tested using the dataset images. In this system we will integrate it with a user interface using flask.
3	Novelty / Uniqueness	Here we provide the result of the prediction input with the infected level stages of disease and also detect the early prediction of the disease.

4	Social Impact / Customer Satisfaction	This model can detect the level of diabetic retinopathy from early-to-late stages.
5	Scalability of the Solution	This system model is probably more cost efficient than other screening test of the diabetic retinopathy. Also, the model is scalable from the architecture and dataset training perspective.

PROBLEM SOLUTION FIT:

Project Title: Deep Learning Fundus Image Analysis for Early Detection of Diabetic retinopathy.

Project Design Phase I - Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

The early detection important for the diabetic patients as diabetic retinopathy is irreversible. The Diabetic retinopathy can be detected using the fundus image of the patient and can be stored in the database.This is more useful than the manual examination

6. CUSTOMER CONSTRAINTS CC

The diabetic retinopathy does not nave any specific symptoms so they fail to notice the illness. Many people do not know about diabetic retinopathy and its adverse reaction.

5. AVAILABLE SOLUTIONS AS

Laser treatment to treat the growth of new blood vessels at the back of the eye (retina) in cases of proliferative diabetic retinopathy, and to stabilize some cases of maculopathy. eye injections - to treat severe maculopathy that's threatening your sight.

Explore AS, differentiate

Focus on J&P, tap into BE understand RC

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

The problem is once the diabetic retinopathy is severe, it cannot be done.
And the severity of diabetic retinopathy results in serious eye illness and also results in losing vision. So, the early detection is important if the patient has diabetes.

9. PROBLEM ROOT CAUSE RC

Diabetic retinopathy is caused by changes in the blood vessels of the retina, the light-sensitive layer of tissue at the back of the inner eye. In some people with diabetic retinopathy, the blood vessels in the retina may swell and leak fluid. In others, abnormal new blood vessels grow on the surface of the retina.

7. BEHAVIOUR BE

This model helps in the early detection of diabetic retinopathy using the fundus images.
It consumes less time than the manual examination.
Also, accuracy is more compared to other techniques.

Focus on J&P, tap into BE understand RC

Identify Strong TR & EM

3. TRIGGERS TR

The triggers in diabetic retinopathy patients are
Spots or dark strings floating in your vision (floaters) Blurred vision.
Fluctuating vision.
Dark or empty areas in vision.

vision loss.

4. EMOTIONS: BEFORE / AFTER EM

Before: Adverse emotional responses include fear, anxiety. vulnerability, guilt, loss of confidence, anger, stress and self perception issues.
After: Early detection and diagnosis gives sense of hope among patients.

10. YOUR SOLUTION SL

Our solution involves the deep learning model with fundus images that detect the severity of the diabetic retinopathy among diabetic patients and the apt diagnosis done after the early detection.

8. CHANNELS OF BEHAVIOUR CH

The diabetic patients have to take the eye examination in the regular interval time. Then only retinopathy can be detected early and proper diagnosis can be done.

REQUIREMENT ANALYSIS:

FUNCTIONAL REQUIREMENT:

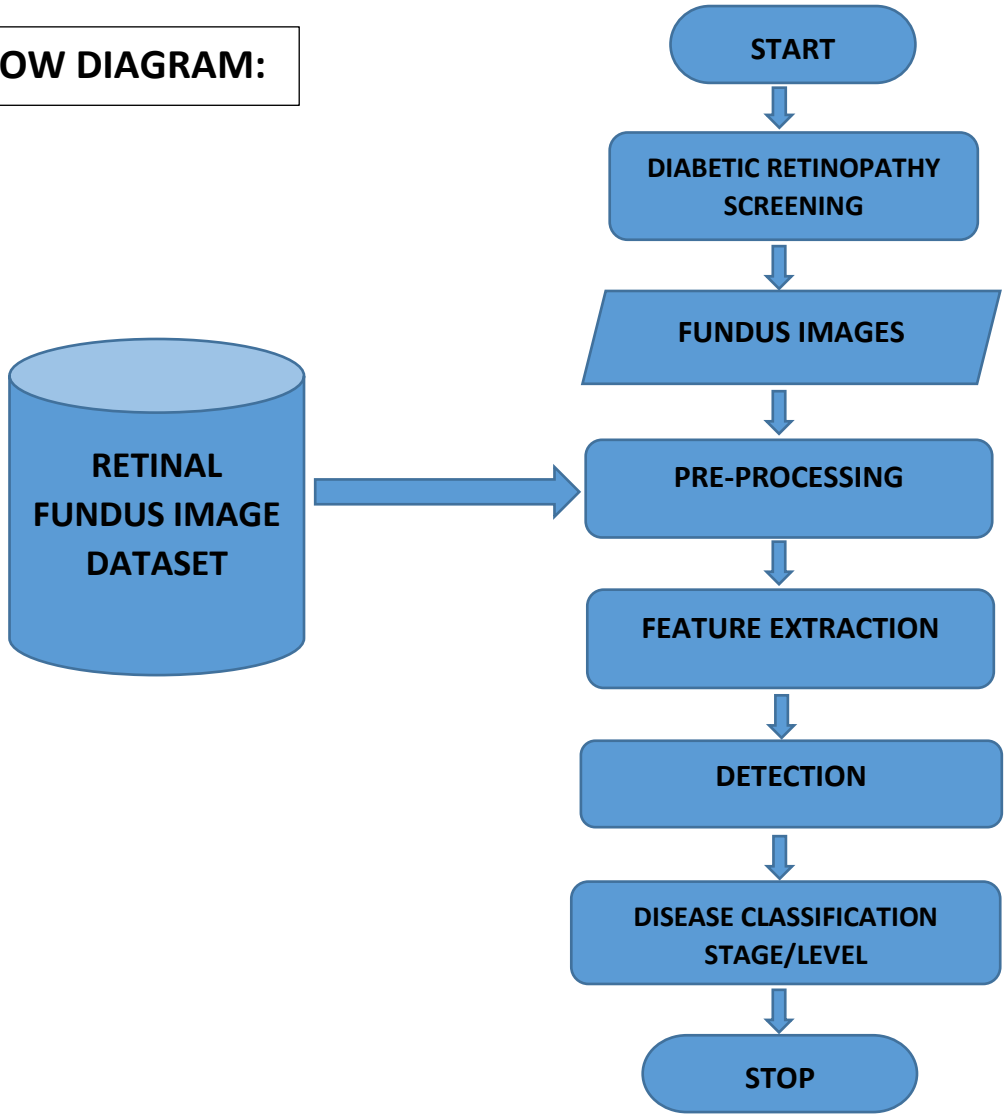
FR.NO	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via OTP Confirmation via mail
FR-3	Input Specification	Fundus image of retina of the patients
redFR-4	Diagnosis of Diabetic Retinopathy	The Diagnostic must be accurate and the training model must have an accuracy over 90% (greater than current manual method)
FR-5	Prediction process	Classification of the input image using the trained deep learning model which is already connected with the web UI
FR-6	Output Specification	Output with all clinical grades of the diabetic retinopathy from early-to-late stages

NON-FUNCTIONAL REQUIREMENT:

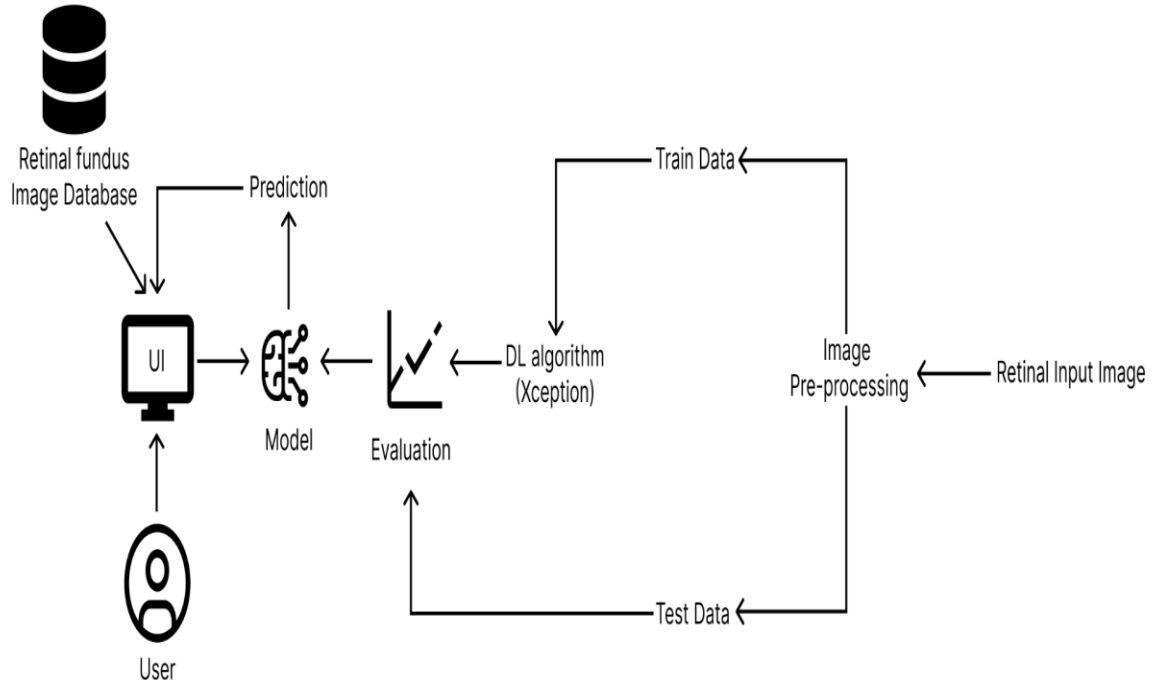
NFR-NO	Non-Functional Requirement	Description
NFR-1	Usability	Easy to use and easy to navigate web service.
NFR-2	Security	Data security is important to store the customer data in the secured manner. The information should not be leaked outside
NFR-3	Reliability	There is a chance of hardware failure 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-4	Performance	The output specifies the stage of disease and the model will provide specific output even for the early stage input image.
NFR-5	Availability	Healthcare affordability, standards, and accessibility is made much more easier using this platform and the application will be available to all kinds of users.
NFR-6	Scalability	The product must hold stable even when multiple users are using at the same times.

PROJECT DESIGN:

DATA FLOW DIAGRAM:



SOLUTION AND TECHNICAL ARCHITECTURE:



PROJECT PLANING AND SCHEDULING:

Sprint Planning And Estimation:

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	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	10	High	KIRTHIK S HARI PRIYA C OVIYA R OYNDRISHA RAKSHIT
Sprint-1	E-mail confirmation	USN-2	As a user, I will receive a confirmation email once I have registered for the application	10	Medium	KIRTHIK S HARI PRIYA C OVIYA R OYNDRISHA RAKSHIT
Sprint-2	Login	USN-3	As a user, I can log into the application by entering my email & password	5	High	KIRTHIK S OYNDRISHA RAKSHIT
Sprint-2	Upload Images	USN-4	As a user,I should be able to upload the image of ECG.	10	High	KIRTHIK S OYNDRISHA RAKSHIT
Sprint-2	Dashboard	USN-5	As a user, based on my requirement I can navigate through the dashboard.	5	Medium	KIRTHIK S HARI PRIYA C OVIYA R OYNDRISHA RAKSHIT
Sprint-3	Train the model	Task 1	As a developer, the dataset will be uploaded and trained by developed algorithm.	20	High	HARI PRIYA C OVIYA R
Sprint-4	Testing & Evaluation	Task 2	As a developer, we tested the trained model using the provided dataset and model will be evaluated for accurate results.	10	High	KIRTHIK S HARI PRIYA C OVIYA R OYNDRISHA RAKSHIT
Sprint-4	Display predicted result	USN-6	As a user, I can view the predicted result in the dashboard.	10	High	KIRTHIK S HARI PRIYA C OVIYA R OYNDRISHA RAKSHIT

Sprint Deliver 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

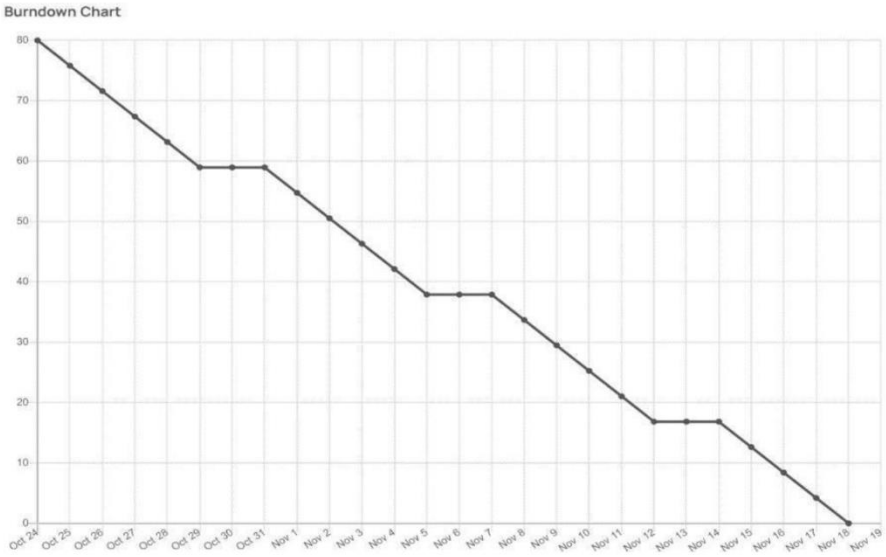
Velocity:

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)

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

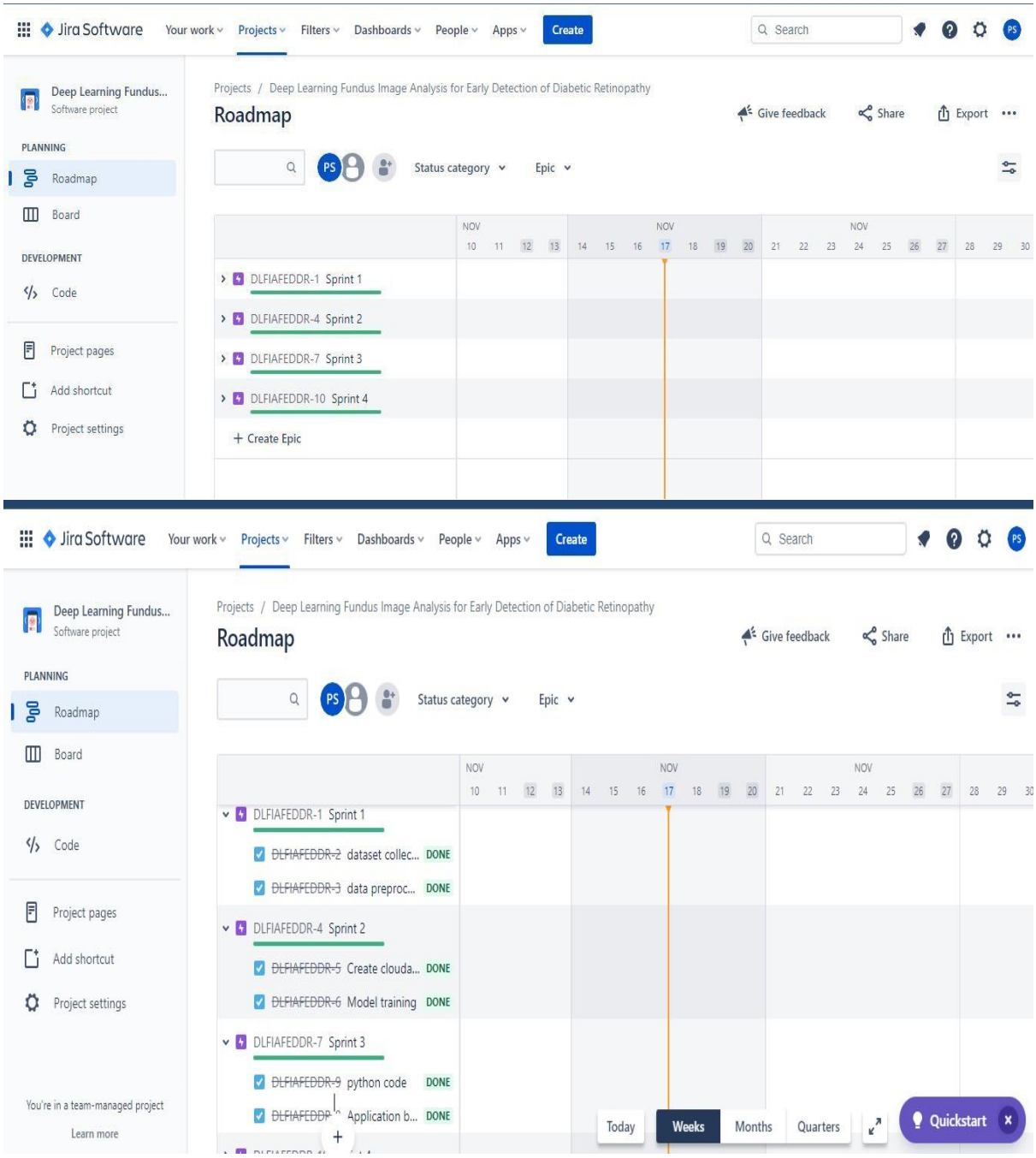
AV=20/6=3.33points per day

Burndown Chart:



A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

JIRA SCREENSHOT:



JIRA Folder is created to show the Scrum methodologies and Burn Down chart progress.

CODING AND SOLUTIONING:

Features 1:

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

Features 2:

We have developed a multilayer deep convolutional neural network that classifies the user image of a eye to which expense 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 developed a messaging service for receiving message for the type of diabetics.

TESTING:

USER ACCEPTANCE TESTING:

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.

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	Severity 4	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

Test-Case Analysis:

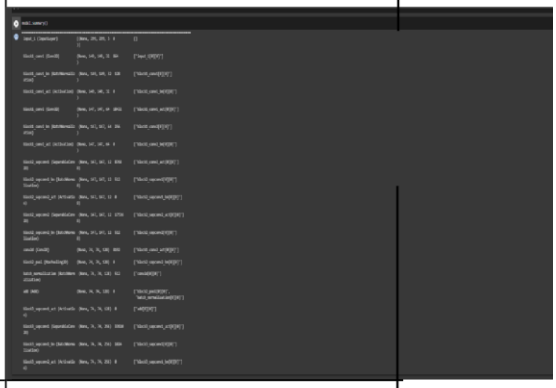
This report shows the number of test cases that have passed, failed, and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	9	0	0	9
Client Application	45	0	0	45
Security	2	0	0	2
Out-source 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:

Model Performance Testing:

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 – **0.7917** Validation Accuracy – **loss 3.2610**	
3.	Confidence Score(Only Yolo Projects)	Class Detected - Confide nce Score -	- - - - - - - -

ADVANTAGES AND DISADVANTAGES:

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.

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.

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.

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:

```
*{
            margin: 0;
            padding: 0;
            font-family: sans-serif;
        }
        .hero{
            height: 100%;
            width: 100%;
            background-image: linear-gradient(to right , #9c27b0, #8ecdff);
            background-position: center;
            background-size: cover;
            position: absolute;
        }
        nav{
            width: 80%;
            position: sticky;
            margin: 20px auto  ;
            z-index: 1;
            display: flex;
            align-items: center;
            }

             .logo img {
            width: 450px;
            }


        .nav-links{
            flex: 1;
```

```
    text-align: right;
}
.nav-links ul li{
    list-style: none;
    display: inline-block;
    margin: 0 20px;
}
.nav-links ul li a{
    color: #FFE5B4;
    text-decoration: none ;
}

.form-box{
    width: 380px;
    height: 480px;
    position: relative;
    margin-top: 80px;
    margin-left:650px;

    background: transparent;
    padding: 5px;
    border-radius: 30px;
    overflow: hidden;
    align: center;
}
.button-box{
    width: 220px;
    margin: 35px auto;
    position: relative;
    box-shadow: 0 0 20px 9px #ff61241f;
    border-radius: 30px;
}
.toggle-btn{
    padding: 10px 30px;
    cursor: pointer;
    background: transparent;
    border: 0;
    outline: none;
    position: relative;
}
#btn{
    top: 0;
    left: 0;
    position: absolute;
    width: 110px;
     height: 100%;
```

```
     background: linear-gradient(to right, #ff105f,#ffad06);
     border-radius: 30px;
     transition: .5s;
}
.social-icons{
    margin: 30px auto;
    text-align: center;
}
.social-icons img{
    width: 30px;
    margin: 0 12px;
    box-shadow: 0 0 20px 0 #7f7f7f3d;
    cursor: pointer;
    border-radius: 50%;
}
.input-group{
    top: 180px;
    position: absolute;
    width: 280px;
    transition: .5s;
}
.input-field{
     width: 100%;
    background: transparent;
    border: 1px solid #fff;
    margin: 6px 0;
    height: 32px;
    border-radius: 20px;
    top: 15px;
    padding: 0 10px;
    box-sizing: border-box;
    outline: none;
    text-align: center;
    color: #fff;
}
::placeholder{
    color: #FFE5B4;
    font-size: 12px;
    }
.submit-btn{
    width: 85%;
    padding: 10px 30px;
    cursor: pointer;
    display: block;
    margin: auto;
    background: linear-gradient(to right, #ff105f,#ffad06);
```

```
    border: 0;
    outline: none;
    border-radius: 30px;
}
.check-box{
    margin: 30px 10px 30px 0;
    }
span{
    color: #fff;
    font-size: 12px;
    bottom: 68px;
    position: absolute;
}
#Login{
    left: 50px;
}
#Register{
    left: 450px;
}

.information{
    width: 1000px;
    height: 1000px;
    position: absolute;
    top: 50%;
    left:-10%;
    transform: translateY(-50%);
}
.information img{
width:400px;
left: 150px;
top: 520px;
}

.eye{
    width:80px;
    position: absolute;
    top: 50%;
    left: 35%;
    transform: translateY(-50%);
    z-index: 1;
}
.overlay{
    width: 0;
    height: 0;
    border-top: 500px solid #fff;
```

```
    border-right: 500px solid transparent;
    border-bottom: 500px solid #fff;
    border-left: 500px solid #fff;
}

 <!DOCTYPE
 html>
             <html lang="en">
             <head>
                 <meta charset="UTF-8">
                 <title>index</title>
                 <link rel="stylesheet" href="..\static\css\indexstyle.css">
             </head>
             <body>
                 <div class="main">
                     <nav>
                             <div class ="logo">
                                 <img src="..\static\images\img.png">
                             </div>
                             <div class="nav-links">
                                 <ul>
                                     <li><a href="index.html">Home</a></li>
                                     <li><a href="login">Login</a></li>
                                     <li><a href="register">Register</a></li>
                                     <li><a
             href="prediction">Prediction</a></li>
                                 </ul>
                             </div>
                     </nav>

                     <div class="information">
                         <div class="overlay"></div>
                         <img src="..\static\images\myproject.jpg"
             class="eye">
                         <div id="circle">
                             <div class="stages one">
                                 <img src="..\static\images\text-1.png">
                                 <div>
                                     <h1>NPDR</h1>
                                     <p>Mild Swelling in Retinal!</p>
                                 </div>
                             </div>
                             <div class="stages two">
                                 <img src="..\static\images\text-2.png">
                                 <div>
                                     <h1>Moderate NPDR</h1>
```

```
                            <p>NPDR plus small bleeds in Retinal!</p>
                        </div>
                    </div>
                    <div class="stages three">
                        <img src="..\static\images\text-3.png">
                        <div>
                            <h1>Severe NPDR</h1>
                            <p>Moderate NPDR plus further damage
to<br>
                                blood vessels in Retinal!</p>
                        </div>
                    </div>
                    <div class="stages four">
                        <img src="..\static\images\text-4.png">
                        <div>
                            <h1>PDR</h1>
                            <p>Severe NPDR plus new Vessel<br>
                                formation in Retinal!</p>
                        </div>
                    </div>
                </div>
            </div>
            <div class="coltrols">
                <img src="..\static\images\arrow.png" id="upBtn">
                <h3>Different Stages of DR</h3>
                <img src="..\static\images\arrow.png" id="downBtn">
            </div>

        </div>
        <script>
            var circle = document.getElementById("circle");
            var upBtn = document.getElementById("upBtn");
            var downBtn = document.getElementById("downBtn");

            var rotateValue = circle.style.transform;
            var rotateSum;

            upBtn.onclick = function()
            {
                rotateSum = rotateValue + "rotate(-90deg)";
                circle.style.transform = rotateSum;
                rotateValue = rotateSum;
            }
             downBtn.onclick = function()
            {
                rotateSum = rotateValue + "rotate(90deg)";
```

```
            circle.style.transform = rotateSum;
            rotateValue = rotateSum;
        }
    </script>

</body>
</html>

 *{
            margin: 0;
            padding: 0;
            font-family: sans-serif;
        }
        .main{
            width: 100%;
            height: 100vh;
            position: relative;
            overflow: hidden;
            background: linear-gradient(to right , #9c27b0,
        #8ecdff);
        }
        nav{
            width: 80%;
            position: sticky;
            margin: 20px auto  ;
            z-index: 1;
            display: flex;
            align-items: center;
            }


            .logo img {
            width: 450px;
            }


            .nav-links{
            flex: 1;
            text-align: right;
        }
        .nav-links ul li{
            list-style: none;
            display: inline-block;
            margin: 0 20px;
        }
        .nav-links ul li a{
```

```
    color: #FFE5B4;
    text-decoration: none ;
}
.content h1{
    font-size: 50px;
    margin-top: 250px;
    margin-left: 550px;
    font-style: italic;
    color: white;
    font-weight: bolder;
    color: #fff;
}
.content p{
    margin: 20px auto;
    font-weight: 200;
    line-height: 25px;
    font-size: 20px;
    margin-left: 250px;
    text-align: center;
    color:#FFE5B4 ;
    }

.information{
    width: 1000px;
    height: 1000px;
    position: absolute;
    top: 50%;
    left:-10%;
    transform: translateY(-50%);
}
.information img{
width:400px;
left: 150px;
top: 520px;
}

.eye{
    width:80px;
    position: absolute;
    top: 50%;
    left: 35%;
    transform: translateY(-50%);
    z-index: 1;
}
.overlay{
    width: 0;
```

```
    height: 0;
    border-top: 500px solid #fff;
    border-right: 500px solid transparent;
    border-bottom: 500px solid #fff;
    border-left: 500px solid #fff;
}

 *{
            margin: 0;
            padding: 0;
            font-family: sans-serif;
        }

        .main{
            width: 100%;
            height: 100vh;
            position: relative;
            overflow: hidden;
            background: linear-gradient(to right ,
        #9c27b0, #8ecdff);
        }
        nav{
            width: 80%;
            position: sticky;
            margin: 20px auto  ;
            z-index: 1;
            display: flex;
            align-items: center;
        }


        .logo img {
            width: 550px;

        }
        .nav-links{
            flex: 1;
            text-align: right;
        }
        .nav-links ul li{
            list-style: none;
            display: inline-block;
            margin: 0 20px;
        }
        .nav-links ul li a{
            color: #FFE5B4;
```

```
    text-decoration: none ;
    margin-top: -10px;
}
#prediction .prediction-input{
    display: flex;
    align-items: center;
    justify-content: center;
    margin-top: 1.5rem;
}
#prediction .prediction-input form{
    margin-left: 1.2rem;
}
#prediction .circle {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    margin-bottom: 5px;
    box-shadow:var(--box-shadow);
    transition:all ease-in 1s;
}

.output{
    width: 200px;
    margin: 10rem 1.5rem;
    padding: 6px;
    text-align: center;
    box-shadow: rgba(0, 0, 0, 0.35) 0px 5px
15px;
}
.output-container{
    display: grid;
    row-gap: 10px;
    grid-template-areas: 'img1 img2 img3
img4 img5 img6';
}

/* Hidden class */
.hidden{
    visibility: hidden;
}
.hide{
    visibility: hidden;
}
.line img{
width: 500px;
position: relative;
```

```
margin-left: 400px;
}

.content h5{
    font-size: 15px;
    margin-left: 400px;
    font-style: italic;
    color: white;
    font-weight: bolder;
    color: #fff;

}
```

PYTHON NOTEBOOK SCREENSHOTS:

DATA COLLECTION

Clonning kaggle in google colab

In [1]:
```
! pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (1.5.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from kaggle) (4.64.1)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from kaggle) (1.24.3)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (from kaggle) (1.15.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packages (from kaggle) (6.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from kaggle) (2.23.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from kaggle) (2022.9.24)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->kaggle) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->kaggle) (3.0.4)
```

In [5]:
```
from google.colab import files
files.upload()
```

```
Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving kaggle.json to kaggle.json
```

Out[5]:
```
{'kaggle.json': b'{"username":"thejeshravikumar","key":"531cf5b6e4fc7ef5ae986e835ff864e1"}'}
```

In [6]:
```
! mkdir ~/.kaggle
```

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

In [7]:
```
! cp kaggle.json ~/.kaggle/
```

In [8]:
```
! chmod 600 ~/.kaggle/kaggle.json
```

Downloading the Dataset

In [11]:
```
!kaggle datasets download -d arbethi/diabetic-retinopathy-level-detection
```

```
Downloading diabetic-retinopathy-level-detection.zip to /content
100% 9.65G/9.66G [05:19<00:00, 32.6MB/s]
100% 9.66G/9.66G [05:19<00:00, 32.4MB/s]
```

Unzipping the Dataset

In [12]:
```
! 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
```

CREATE DATABASE

In [30]:
```
!pip install -q cloudant
```

```
     |████████████████████████████████| 80 kB 4.0 MB/s
```

In [31]:
```
from cloudant.client import Cloudant
```

In [32]:
```
# Authenticate using an IAM API key

client = Cloudant.iam('**********', '**********', connect=True)
```

In [33]:
```
#Create a database using an initialized client

my_database = client.create_database('my_database')
```

Model Building

In [48]:
```
xception = Xception(input_shape=imageSize + [3], weights='imagenet', include_top=False)
```

In [49]:
```
for layer in xception.layers:
  layer.trainable = False
```

In [50]:
```
x = Flatten()(xception.output)
```

In [51]:
```
prediction = Dense(5, activation='softmax')(x)
```

In [52]:
```
model = Model(inputs=xception.input, outputs=prediction)
```

In [53]:
```
model.summary()
```

In [18]:
```
#install kaggle
!pip install -q kaggle
```

In [19]:
```
from google.colab import files

files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle (1).json
```

Out[19]:
```
{'kaggle.json': b'{"username":"venishkaran","key":"6a5541e3944fdbfbb06760e6f6e220f9"}'}
```

In [20]:
```
#create a kaggle folder
!mkdir ~/.kaggle
```

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

In [21]:
```
#copy the kaggle.json file into kaggle folder
!cp kaggle.json ~/.kaggle/
```

In [22]:
```
# permission for the json to act
!chmod 600 ~/.kaggle/kaggle.json
```

In [23]:
```
!kaggle datasets list
```

```
ref                                                         title                                size  lastUpdated          downloadCount
voteCount  usabilityRating
----------------------------------------------------------  -----------------------------------  ------  -------------------  -------------
---------  --------------
akshaydattatraykhare/diabetes-dataset                       Diabetes Dataset                       9KB  2022-10-06 08:55:25           9583
311  1.0
whenamancodes/covid-19-coronavirus-pandemic-dataset         COVID -19 Coronavirus Pandemic Dataset    11KB  2022-09-30 04:05:11           7684
245  1.0
```

DATA COLLECTION

Clonning kaggle in google colab

In [1]:
```
! pip install kaggle
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: kaggle in /usr/local/lib/python3.7/dist-packages (1.5.12)
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from kaggle) (4.64.1)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages (from kaggle) (1.24.3)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.7/dist-packages (from kaggle) (1.15.0)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.7/dist-packages (from kaggle) (6.1.2)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from kaggle) (2.23.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from kaggle) (2022.9.24)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.7/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->kaggle) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->kaggle) (3.0.4)
```

In [5]:
```
from google.colab import files
files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle.json
```

Out[5]:
```
{'kaggle.json': b'{"username":"thejeshravikumar","key":"531cf5b6e4fc7ef5ae986e835ff864e1"}'}
```

In [6]:
```
! mkdir ~/.kaggle
```

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

In [7]:
```
! cp kaggle.json ~/.kaggle/
```

In [8]:
```
! chmod 600 ~/.kaggle/kaggle.json
```

In [10]:
```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [11]:
```
r = model.fit(training_set, validation_data=test_set, epochs=30, steps_per_epoch=len(training_set)//32, validation_steps=len(test_set)//32)
```

```
Epoch 1/30
3/3 [==============================] - 52s 5s/step - loss: 10.3830 - accuracy: 0.2917
Epoch 2/30
3/3 [==============================] - 26s 8s/step - loss: 20.3412 - accuracy: 0.5521
Epoch 3/30
3/3 [==============================] - 24s 8s/step - loss: 15.4075 - accuracy: 0.6042
Epoch 4/30
3/3 [==============================] - 19s 6s/step - loss: 8.4522 - accuracy: 0.4271
Epoch 5/30
3/3 [==============================] - 20s 6s/step - loss: 7.6308 - accuracy: 0.4688
Epoch 6/30
3/3 [==============================] - 19s 5s/step - loss: 4.7706 - accuracy: 0.6146
Epoch 7/30
3/3 [==============================] - 16s 5s/step - loss: 5.9880 - accuracy: 0.6667
Epoch 8/30
3/3 [==============================] - 15s 5s/step - loss: 6.4603 - accuracy: 0.6667
Epoch 9/30
3/3 [==============================] - 12s 4s/step - loss: 4.2353 - accuracy: 0.7396
Epoch 10/30
3/3 [==============================] - 15s 4s/step - loss: 4.3710 - accuracy: 0.6979
Epoch 11/30
3/3 [==============================] - 13s 4s/step - loss: 4.4179 - accuracy: 0.6562
Epoch 12/30
3/3 [==============================] - 13s 4s/step - loss: 5.3877 - accuracy: 0.5625
Epoch 13/30
3/3 [==============================] - 14s 4s/step - loss: 6.7465 - accuracy: 0.7083
Epoch 14/30
3/3 [==============================] - 13s 4s/step - loss: 5.9678 - accuracy: 0.7188
```

In [1]:
```
import tensorflow as tf
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
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 [2]:
```
imageSize = [299, 299]
```

In [3]:
```
#importing data set
train_datagen = ImageDataGenerator(rescale=1/255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen = ImageDataGenerator(rescale=1/255)
```

In [4]:
```
training_set = train_datagen.flow_from_directory(
    "dataset\\preprocessed dataset\\preprocessed dataset\\training", target_size = (299, 299), batch_size=32, class_mode = 'categorical')

test_set = train_datagen.flow_from_directory(
    "dataset\\preprocessed dataset\\preprocessed dataset\\testing", target_size=(299, 299), batch_size=32, class_mode='categorical')
```

```
Found 3662 images belonging to 5 classes.
Found 734 images belonging to 5 classes.
```

In [5]:
```
xception = Xception(input_shape=imageSize + [3], weights='imagenet', include_top=False)
```

In [6]:
```
for layer in xception.layers:
    layer.trainable = False
```

In [7]:
```
x = Flatten()(xception.output)
```

In [18]:
```
#install kaggle
!pip install -q kaggle
```

In [19]:
```
from google.colab import files

files.upload()
```

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving kaggle.json to kaggle (1).json
```

Out[19]:
```
{'kaggle.json': b'{"username":"venishkaran","key":"6a5541e3944fdbfbb06760e6f6e220f9"}'}
```

In [20]:
```
#create a kaggle folder
!mkdir ~/.kaggle
```

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

In [21]:
```
#copy the kaggle.json file into kaggle folder
!cp kaggle.json ~/.kaggle/
```

In [22]:
```
# permission for the json to act
!chmod 600 ~/.kaggle/kaggle.json
```

In [23]:
```
!kaggle datasets list
```

```
ref                                                         title                                   size  lastUpdated          downloadCount
voteCount  usabilityRating
----------------------------------------------------------  --------------------------------------  ------  -------------------  -------------
---------  --------------
akshaydattatraykhare/diabetes-dataset                       Diabetes Dataset                         9KB  2022-10-06 08:55:25           9583
311  1.0
whenamancodes/covid-19-coronavirus-pandemic-dataset         COVID -19 Coronavirus Pandemic Dataset  11KB  2022-09-30 04:05:11           7684
245  1.0
```

```
@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 successsful, 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)
```

```
from flask import Flask, render_template, request, redirect, url_for
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 cloudant.client import Cloudant

# basepath = os.path.dirname(__file__)
# print(basepath)
print(os.getcwd())
model = load_model(r'updated_Xception.h5')
client = Cloudant.iam("ca414387-f653-4fa8-8a90-a828be636391-bluemix",
                      "OSs0X0E0p-9BoyBZcMpe2sgf8h8gAJMUmYyNKGst9LIy", 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')




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


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

GITHUB LINK: https://github.com/IBM-EPBL/IBM-Project-43030-1660712110

DEMO LINK: https://photos.app.goo.gl/cyk8RnVEEryi8A3R7