PROJECT REPORT

DeepLearning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

1.INTRODUCTION :

1.1 Project Overview :

The problem statement of this project is a disease that is caused by Diabetics. 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. Diabetic Retinopathy 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.The evaluation of diabetic retinopathy is associated with peoples having diabetics.The evaluation will be based on the fundus or retinal images of the diabetic patients eye.In project will be best for the diabetic patients for the earlier detection of diabetic retinopathy. 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 main purpose of this project is to help the diabetic patients for the early detection of diabetic retinopathy.The root cause of the diabetic retinopathy is because of high sugar level in the blood due to the diabetics.One of the main cause of diabetic retinopathy is people fail to notice the illness and that cause the adverse reaction .This project will help them to detect diabetic retinopathy at the earlier stage and it can be treated easily.

As diabetic retinopathy progresses it blocks the tiny blood vessels that nourish the retina andcut off its blood supply.This project will help to detect diabetic retinopathy at the early stage by analysing Fundus images.This will provide the result with better accuracy and saves the time and cost of the patient. This will helps the patient to recover from the diabetic retinopathy in a better way.

Diabetics patients are not aware of the complications of the diabetics so they fail to notice these serious diseases.Diabetic retinopathy doesn't have any specific symptoms other than blurred vision so many people will fail to notice the illness and the adverse reaction of the diabetic retinopathy.This project will help them for the early detection of Diabetic Retinopathy.

2.LITERATURE SURVEY

2.1 Existing problem:

Diabetic Retinopathy (DB) is a complication of diabetes that influences the eyes. Damage to blood vessels in the tissue of the retina, the back layer of the eye, typically causes itBlurriness, floaters, dark or empty areas in the vision, and difficulty recognizing color blindness are some of the early symptoms. It necessitates constant monitoring, and in the event of complications, it may shorten life expectancy. If it is not diagnosed and treated, it can blind you. The medication cannot be cured at this time. Diabetic retinopathy can be stopped or slowed down with treatment.

Diabetes management may be used carefully to treatmild cases.

The diabetes on a fundus image is identified by the proposed method, which makes use of an Alex net Convolutional Neural Network (CNN).The dataset that was used came from the
MESSIDOR database. It has 1200 images of the fundus and wasdivided into 580 images of normal and exudates for the project. The dataset has been divided into two parts for the CNN
process: the training dataset and the testing dataset. On 50% of the training dataset, this method achieves accuracy greater than 90%, and the remaining 50% of the dataset is used for
testing. The tests give an accuracy of about 85%. Even though the images received a good accuracy, only 580 were utilized for both training and testing, despite the fact that the dataset was insufficient to train the neural network. Additionally, it had trouble identifying the image's smaller exudates.

In order to categorize diabetic retinopathy in the fundus imagery into five categories—No DR, Mild DR, Moderate DR, Severe DR, and Proliferative DR—the proposed system developed a CNN architecture. They have examined previous efforts to detect DR using CNN, and they have altered the networks in CNN to improve its accuracy and efficiency. They have achieved a 75% accuracy on the dataset of 80000 images. Classification of DR into mild, moderate, and severe forms presents some challenges.

To analyze the fundus image and predict the stage, they used a Deep Convolutional Neural Network (DCNN), which includes No DR, Moderate DR (a combination of mild and moderate Non-Proliferative DR), and Severe DR (severe NPDR and Proliferative DR).Over a period of time, they have almost used 3468 fundus images from various Kaggle clinics.They have achieved an accuracy rate of over 80%. When a model is trained with a small dataset and fails when applied to a new dataset, an overfitting problem occurs.

The architecture used in the proposed model isDenseNet121.This is unique in that each feature map output from a convolution layer is concatenated with the subsequent layers of the same block. Based on the severity of the disease, it divides DR into five categories: PDR, No DR, Slight DR, Medium DR, and Severe DR. Cross-testing two datasets—Messidor and APTOS—has been used in the proposed method to enable the model to acquire complex features. They used a cross-testing strategy with unbalanced data, so their accuracy is lower than that of current methods. Additionally, the model had trouble categorizing the Slight NDPR class.

2.2 References:

1] Johari, Mohamad; Hassan, Hiron; Yassin, Ahmad; Tahir, Noorita; Azlee Zabidi; Rizman, Zairi; Baharom, R.; and Wahab, N.Utilizing a deep learning neural network for the early detection of diabetic retinopathy.United Arab Emirates International Journal of Engineering and Technology7.198 10.14419/ijet.v7i4.11.20804

[2] Convolutional Neural Networks for Diabetic Retinopathy, by Harry Pratta, Frans Coenenb, Deborah M Broadbent, Simon P Hardinga, and Yalin Zhenga (2016).
https://doi.org/10.1016/j.procs.2016.07.014.

[3] M. Shaban, Z. Ogur, A. Mahmoud, A. Switala, A. Shalaby, and H. Abu Khalifeh, among others2020) A convolutional neural network for diabetic retinopathy screening and staging.

[4] PLoS ONE 15:e0233514.https://doi.org/10.1371/journal.[4] Ayala, A.; pone.0233514T. Ortiz Figueroa;B. Fernandes;Deep Learning Improved Diabetic Retinopathy Detection, Cruz, F.Appl.Sci;.2021, 11, 11970.Doi

:https://doi.org/10.3390/app112411970.

2.3 Problem Statement Definition:

PROBLEM STATEMENT 1: The Physician will examine the medical condition of the eye and try to predict diabetic retinopathy which is manually sternous.

PROBLEM STATEMENT 2: The medical practitioner will determine the diabetic retinopathy and to inspect the medical condition of the patient to measure the extent of diabetic retinopathy.

PROBLEM STATEMENT 3: The Oculist have to suggest the treatment and to measure the level of diabetic retinopathy and provide treatment based on the extent of diabetic retinopathy.

PROBLEM STATEMENT 4: In a serious level the opthamalogist will find the reason for the blurry vision and provide treatment for diabetic retinopathy.

3: IDEATION AND PROPOSED SOLUTION:

3.1 Empathy map canvas:

The empathy map demonstrates the different emotions of the diabetic patients like what do they think and feel,what do they say and do,what do they hear, what to they see ,Pain and gain of the diabetic patients.

What do they think and feel?

Feel depressed,Wishing to have a normal eyesight, irritated eyes and want to maintain a healthy food diet.

What do they hear?

They will hear so many comments like have a regular check up,manage the blood level correctly,maintain proper food diet and take sufficient vitamins.

What do they see?

Encouragements from families and friends,research on more successful solutions and consulting specialized doctors and physicians.

What do they see and do ?

Starts planning their life in a right manner,makes more research on the diseases.They will understand the problem and try to persuade .

3.2 IDEATION AND BRAINSTORMING

The brainstorming is a process of sourcing ideas from the team members and collaborating all the ideas and making a perfect idea. Different ideas were given by different members like the different symptoms of the disease, different algorithms that can be used to train the model,the process of getting the fundus images, the preventing measures of the disease and also the best treatment for the accurate level of the disease. These all ideas were combined and the ideation had done in a best way.

3.3 PROPOSED SOLUTION:

Diabetic Retinopathy is one of the emerging diseases which is the reason for blindness. DR mutilates the retinal blood vessels of a patient having diabetes. Diabetic Retinopathy (DR) is an ophthalmic disease that damages retinal blood vessels. DR causes imperfect vision and may cause blindness if it is notdiagnosed in early stages. Early detection of DiabeticRetinopathy includes the identification of microaneurysms and hemorrhages. Because the signs and symptoms of diabetic
retinopathy are typically not present during the first stage of the disease, it can often go undiagnosed until damage to vision has
occurred. Existing methods are lacking in the earlierdetection.Because preprocessing techniques used in those methods are not effective to analyze such smaller features (nearly 10 microns to 100 microns).

We opt to use multi-layer neural networks as deep NN.
Due to the fact that data is Image, the best type of neural network satisfying our goal is Convolutional Neural Networks.
As we have to do for most of the data, normalization plays an important role in our process. Before doing any tasks, preprocessing images (our dataset) is highlyrecommended.Consequently better accuracy will be
achieved by preprocessed data. After preprocessing and normalizing, the prepared dataset could be used as input to
our deep convolutional neural network. Then deep NN will be run and fit to our data and the result will be produced by that.
This report will cover step by step how this deep convolutional network be implemented.

One of the major decisions had to be made was choosing the suitable programming language satisfying our goal for extracting knowledge from

our data. After some searching the suitable decision has been made by selecting Python as the
project programming language. Due to the fact that, a lot of tools and frameworks are available for Python to create powerful Artificial Neural Networks. Also IBM Watson helps to predict future outcomes,automate complex processes and
optimize user’s time. And also the result accuracy will be
increased from 70% which is the accuracy of the test results that the previous developed codes produced.

It Reduction of Diabetic Retinopathy risk. It Provides Digital Assistance. It is Very helpful in making decisions faster. It Can be used 24x7.

This can be implemented as an essential diagnosis method in every hospital. Accurate detection and analysis can encourage the increase in financial benefit. It can collaborate with the government for health awareness camps

Accurate predictions and extensive use. Based on the times of the correct diagnosis. Availability. This project will help us to detect DR more precisely than the existing methodologies.
Also it can produce a result which specifies the stages of Diabetic Retinopathy.

3.4 PROBLEM SOLUTION FIT:

The evaluation of the Diabetic Retinopathy is associated with peoples having Diabetes.The evaluation will be based on the fundus or retinal images of the diabetic patients eye.This project will be best for the diabetic patients for the earlier detection of diabetic retinopathy.Diabetic retinopathy is one of the serious consequence of diabetics, earlier detection of diabetic retinopathy will help the patients to recover from the disease effectively.Advising Diabetic patients not to intake high level of sugars and to maintain a normal blood pressure and cholesterol in order to prevent them from diabetic retinopathy. Diabetics patients are not aware of the complications of the diabetics so they fail to notice these serious diseases.Diabetic retinopathy doesn't have any specific symptoms other than blurred vision so many people will fail to notice the illness and the adverse reaction of the diabetic retinopathy.The treatments are depend on the severity of the disease.The treatments are mostly focus on slowing or stopping the progression of diabetic retinopathy.There are so many solutions available for diabetic retinopathy some of them are Injecting medications on to the eye,Photocoagulation,Panretinal photocoagulation ,Vitrectomy.Laser treatment is best at treating the growth of new blood vessels.The root cause of the diabetic retinopathy is because of high sugar level in the blood due to the diabetics.One of the main cause of diabetic retinopathy is people fail to notice the illness and that cause the adverse reaction .This project will help them to detect diabetic retinopathy at the earlier stage and it can be treated easily.As diabetic retinopathy progresses it blocks the tiny blood vessels that nourish the retina andcut off its blood supply.This project will help to detect diabetic retinopathy at the early stage by analysing Fundus images.This will provide the result with better accuracy and saves the time and cost of the patient. This will helps the patient to recover from the diabetic retinopathy in a better way.Our solution is that the diabetic patients should aware of the consequence of the diabetics and should monitor their health frequently.And the DEEP LEARNING FUNDUS IMAGE ANALYSIS FOR EARLY DETECTION OF DIABETIC RETINOPATHY will help To diagnose the diabetic retinopathy at the early Stage and it can be easily treated.This model will give them the better accuracy and saves the patient's time and cost.

4: REQUIREMENT ANALYSIS

4.1 FunctionalRequirements:

Following are the functional requirements of the proposed solution.

FRNo.	**Functional Requirement(Epic)**	**SubRequirement(Story/Sub-Task)**
FR-1	Identify and selecting dataset	The appropriate dataset to enhance the model's performance is the necessary to select.
FR-2	Training	It is required to import the libraries needed for the training of the model.
FR-3	Diagnosis	The training should ensure proper diagnosis and make sure to identify the true and false of the medical condition [DiabeticRetinopathy].
FR-4	Analysis	Based on the training the model should analyse the medical condition [DR] in order to predict/detect the disease accurately.
FR-5	Testing	The trained model is tested with different data to ensure that trained well to predict/detect the medical condition[DR].
FR-6	Reporting	The result of the experiment gives the medical report of the disease[DR] so that the patient can understand the level of the disease.

FR-7	Treatment	The testing of the model gives us the level of the medical conditions so that we can go for the required treatment.

4.2 **Non-functionalRequirements:**

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

FRNo.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	User with basic understanding of the medical conditionand computer knowledge canoperatethesystem. User friendlyinterfacethatcanbeaccessedwith easeby users.
NFR-2	**Reliability**	There isachanceofhardwarefailureorfalsepositiveswhenthetestingd ataismoreofdifferent thanthetrainingdataset. Permissiongrantedonlybytheadministratorofthe system
NFR-3	**Performance**	Ifthesystemupdatefailsor bugsinthecodeeventhoughthesystemcanrollbacktoitsinitialstate. Theperformanceofthemodelismeanttogivespeedyresultsforthepat ients.
NFR-4	**Availability**	Thetreatmentshouldbeavailableatlowcostsothateveryonewith DR canfind itbeneficial.
NFR-5	**Scalability**	By processing more datasets for the reference of DR detection.

5. PROJECT DESIGN

5.1: SolutionArchitecture:

Solution architecture is a complex process – with many sub-processes – that bridgesthegapbetweenbusinessproblemsandtechnologysolutions.Itsgoal sareto:

- Findthebesttechsolutiontosolveexistingbusinessproblems.
- Describe the structure, characteristics, behavior, and other aspects of thesoftwaretoprojectstakeholders.
- Definefeatures,developmentphases,andsolutionrequirements.
- Provide specifications according to which the solution is defined, managed,anddelivered.

Technologies needed for Minimum Viable Product deployment

Software technologies required for the systematic development and deployment oftheprojectare:

a. HTML/CSS/JavaScript/bootstrap-FrontEndDevelopment

b. Python

c. TensorFlow

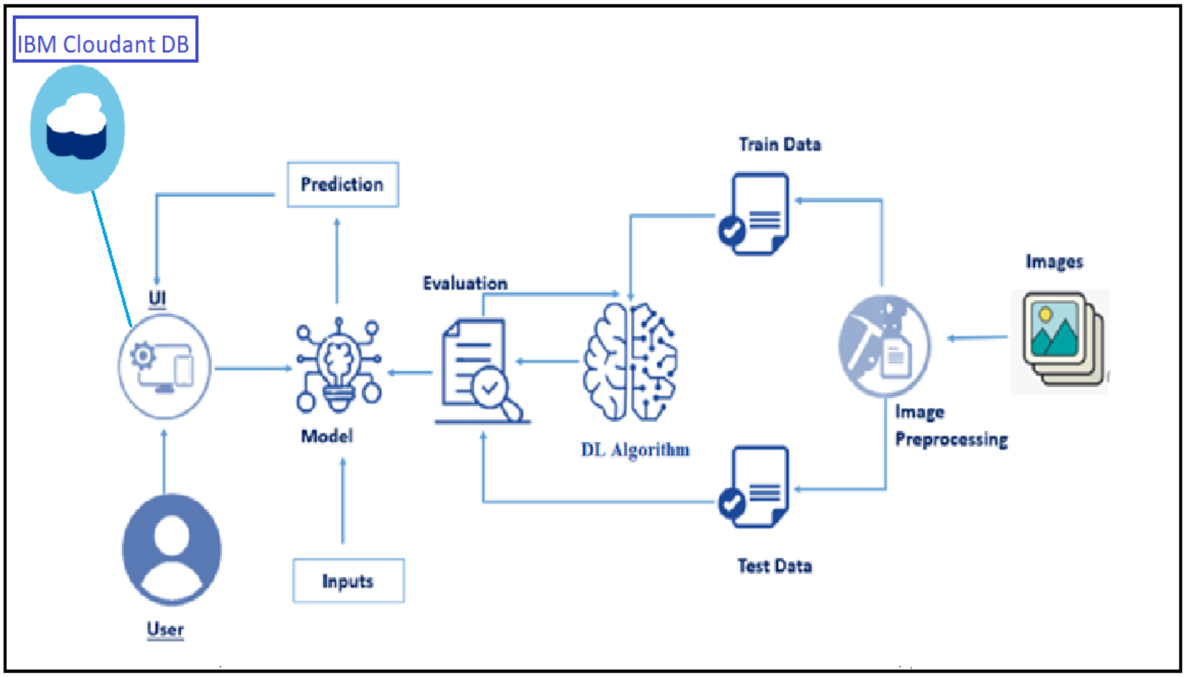
d. ImageprocessingBasics

e. Flask-BackendDevelopment

f. Git&GitHub-projectManagement

g. IBMCloud-Hosting

IBMWatson-TrainingtheDeepLearningModel



6. PROJECT PLANNING & SCHEDULING:

6.1: Sprint Planning & Estimation

To build a DL model we have to split training and testing data into two separate folders. But In the project dataset folder training and testing folders are presented. So, in this case we just have to assign a variable and pass the folder path to it.

Four different transfer learning models are used in our project and the best model (Xception) is selected.

The image input size of xception model is 299, 299.

Data Pre-Processing

import the necessary libraries

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.applications.xception import Xception, preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
```

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

test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
training_set = train_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/training',
                                                 target_size = (299, 299),
                                                 batch_size = 32,
                                                 class_mode = 'categorical')

test_set = test_datagen.flow_from_directory('/content/preprocessed dataset/preprocessed dataset/testing',
                                            target_size = (299, 299),
                                            batch_size = 32,
                                            class_mode = 'categorical')

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

Model Building

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

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/xcep
_kernels_notop.h5
83689472/83683744 [==============================] - 1s 0us/step
83697664/83683744 [==============================] - 1s 0us/step
```

```
# don't train existing weights
for layer in xception.layers:
    layer.trainable = False
```

```
# our layers - you can add more if you want
x = Flatten()(xception.output)
```

```
# view the structure of the model
model.summary()
```

```
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)
```

```
batch_normalization_3 (BatchNo  (None, 10, 10, 1024  4096       ['conv2d_3[0][0]']
rmalization)                   )

add_11 (Add)                   (None, 10, 10, 1024  0          ['block13_pool[0][0]',
                               )                                'batch_normalization_3[0][0]']

block14_sepconv1 (SeparableCon  (None, 10, 10, 1536  1582080    ['add_11[0][0]']
v2D)                           )

block14_sepconv1_bn (BatchNorm  (None, 10, 10, 1536  6144       ['block14_sepconv1[0][0]']
alization)                     )

block14_sepconv1_act (Activati  (None, 10, 10, 1536  0          ['block14_sepconv1_bn[0][0]']
on)                            )

block14_sepconv2 (SeparableCon  (None, 10, 10, 2048  3159552    ['block14_sepconv1_act[0][0]']
v2D)                           )

block14_sepconv2_bn (BatchNorm  (None, 10, 10, 2048  8192       ['block14_sepconv2[0][0]']
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
```

```python
# fit the model
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
)
```

```
Epoch 1/30
3/3 [==============================] - 32s 6s/step - loss: 9.6582 - accuracy: 0.4479
Epoch 2/30
3/3 [==============================] - 15s 5s/step - loss: 9.3711 - accuracy: 0.5417
Epoch 3/30
3/3 [==============================] - 15s 5s/step - loss: 7.4651 - accuracy: 0.5208
Epoch 4/30
3/3 [==============================] - 16s 5s/step - loss: 4.8766 - accuracy: 0.5938
Epoch 5/30
3/3 [==============================] - 15s 5s/step - loss: 6.3676 - accuracy: 0.6458
Epoch 6/30
3/3 [==============================] - 14s 5s/step - loss: 5.2558 - accuracy: 0.6667
Epoch 7/30
3/3 [==============================] - 16s 5s/step - loss: 5.4306 - accuracy: 0.6458
Epoch 8/30
3/3 [==============================] - 13s 4s/step - loss: 4.8280 - accuracy: 0.6562
Epoch 9/30
3/3 [==============================] - 14s 5s/step - loss: 3.9236 - accuracy: 0.6458
Epoch 10/30
3/3 [==============================] - 13s 4s/step - loss: 3.2804 - accuracy: 0.6562
Epoch 11/30
3/3 [==============================] - 15s 5s/step - loss: 2.1550 - accuracy: 0.6875
Epoch 12/30
3/3 [==============================] - 15s 5s/step - loss: 3.0436 - accuracy: 0.6979
Epoch 13/30
3/3 [==============================] - 14s 5s/step - loss: 3.4109 - accuracy: 0.7500
Epoch 14/30
3/3 [==============================] - 15s 5s/step - loss: 2.8810 - accuracy: 0.7396
Epoch 15/30
3/3 [==============================] - 15s 5s/step - loss: 3.4979 - accuracy: 0.6667
Epoch 16/30
3/3 [==============================] - 14s 5s/step - loss: 3.1029 - accuracy: 0.6562
Epoch 17/30
3/3 [==============================] - 14s 5s/step - loss: 2.8477 - accuracy: 0.6979
Epoch 18/30
3/3 [==============================] - 14s 4s/step - loss: 2.6290 - accuracy: 0.6979
Epoch 19/30
3/3 [==============================] - 16s 5s/step - loss: 4.5827 - accuracy: 0.5938
```

```
Epoch 20/30
3/3 [==============================] - 13s 4s/step - loss: 1.7713 - accuracy: 0.7604
Epoch 21/30
3/3 [==============================] - 14s 5s/step - loss: 4.1266 - accuracy: 0.6042
Epoch 22/30
3/3 [==============================] - 15s 5s/step - loss: 2.2045 - accuracy: 0.7188
Epoch 23/30
3/3 [==============================] - 15s 5s/step - loss: 2.7497 - accuracy: 0.7500
Epoch 24/30
3/3 [==============================] - 16s 5s/step - loss: 3.3502 - accuracy: 0.7083
Epoch 25/30
3/3 [==============================] - 15s 5s/step - loss: 3.1592 - accuracy: 0.7188
Epoch 26/30
3/3 [==============================] - 14s 5s/step - loss: 3.2060 - accuracy: 0.6354
Epoch 27/30
3/3 [==============================] - 16s 5s/step - loss: 3.4886 - accuracy: 0.6250
Epoch 28/30
3/3 [==============================] - 15s 5s/step - loss: 3.0558 - accuracy: 0.6979
Epoch 29/30
3/3 [==============================] - 14s 5s/step - loss: 4.7360 - accuracy: 0.6250
Epoch 30/30
3/3 [==============================] - 14s 5s/step - loss: 2.0049 - accuracy: 0.7708
```

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

7. CODING & SOLUTIONING:

Import the necessary libraries.
ImageDataGenerator class is instantiated and the configuration for the types of data augmentation

There are five main types of data augmentation techniques for image data; specifically:

- Image shifts via the width_shift_range and height_shift_range arguments.
- The image flips via the horizontal_flip and vertical_flip arguments.

- Image rotations via the rotation_range argument
- Image brightness via the brightness_range argument.
- Image zoom via the zoom_range argument.

For Training set using flow_from_directory function.

This function will return batches of images from the subdirectories

Arguments:

- directory: Directory where the data is located. If labels are "inferred", it should contain subdirectories, each containing images for a class. Otherwise, the directory structure is ignored.
- batch_size: Size of the batches of data which is 64.
- target_size: Size to resize images after they are read from disk.
- class_mode:
  - ‘int': means that the labels are encoded as integers (e.g. for sparse_categorical_crossentropy loss).
  - 'categorical' means that the labels are encoded as a categorical vector (e.g. for categorical_crossentropy loss).
  - 'binary' means that the labels (there can be only 2) are encoded as float32 scalars with values 0 or 1 (e.g. for binary_crossentropy).
  - None (no labels

For one of the models, we will use it as a simple feature extractor by freezing all the five convolution blocks to make sure their weights don’t get updated after each epoch as we train our own model.

Here, we have considered images of dimension (229,229,3).

Also, we have assigned include_top = False because we are using convolution layer for features extraction and wants to train fully connected layer for our images classification(since it is not the part of Imagenet dataset)

Flatten layer flattens the input. Does not affect the batch size.

A dense layer is a deeply connected neural network layer. It is the most common and frequently used layer.
The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find errors or deviations in the learning process. Keras requires a loss function during the model compilation process.

Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer.

The model is trained for 30 epochs and after every epoch, the current model state is saved if the model has the least loss encountered till that time. We can see that the training loss decreases in almost every epoch till 10 epochs and probably there is further scope to improve the model.

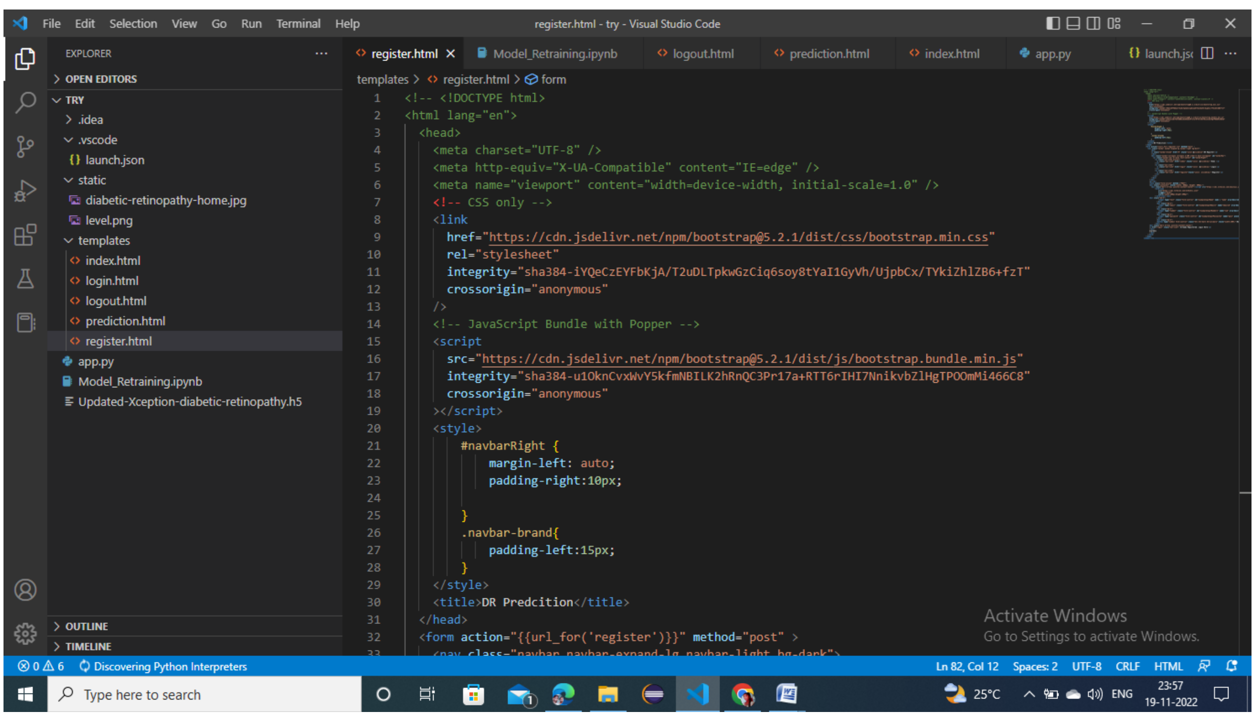
fit_generator functions used to train a deep learning neural network

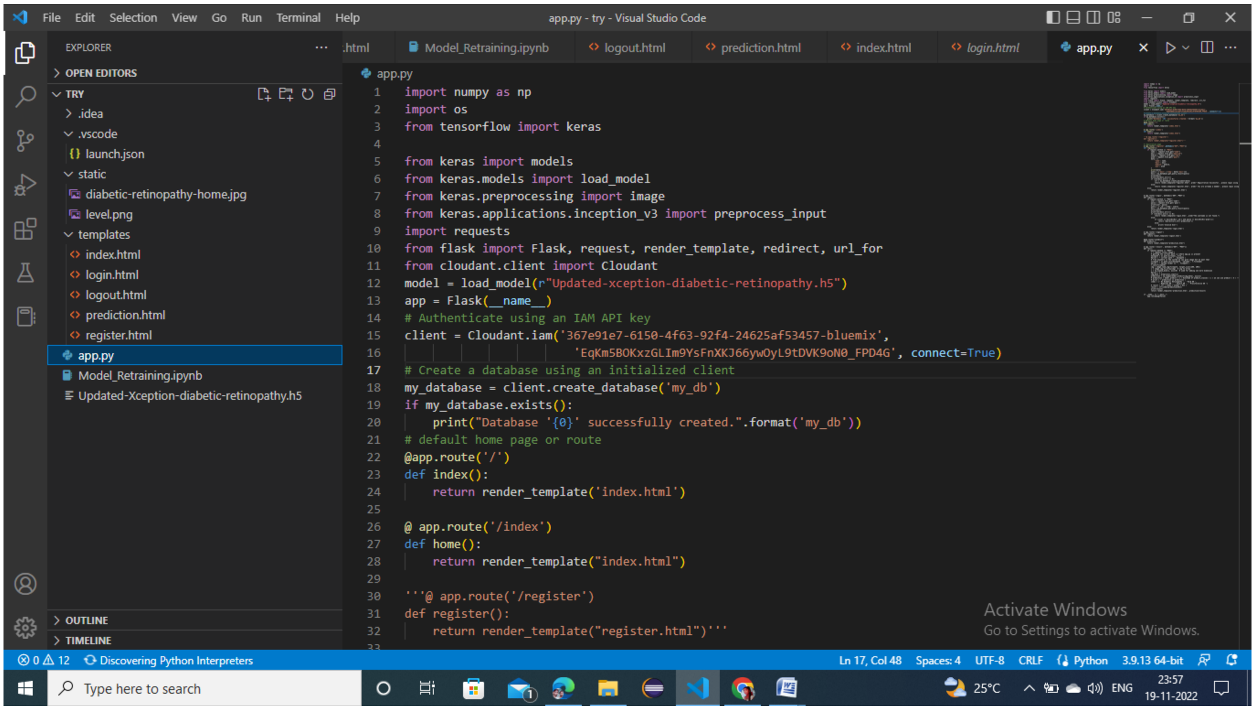
Arguments:

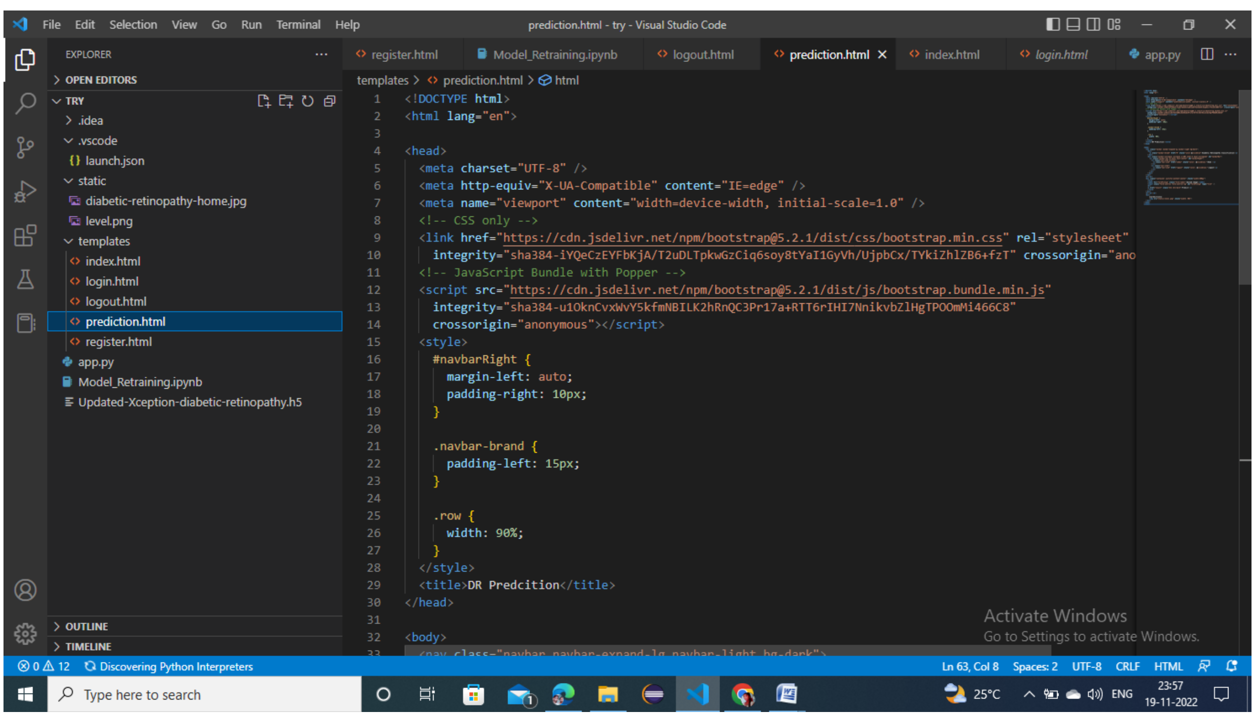
- steps_per_epoch: it specifies the total number of steps taken from the generator as soon as one epoch is finished and the next epoch has started. We can calculate the value of steps_per_epoch as the total number of samples in your dataset divided by the batch size.

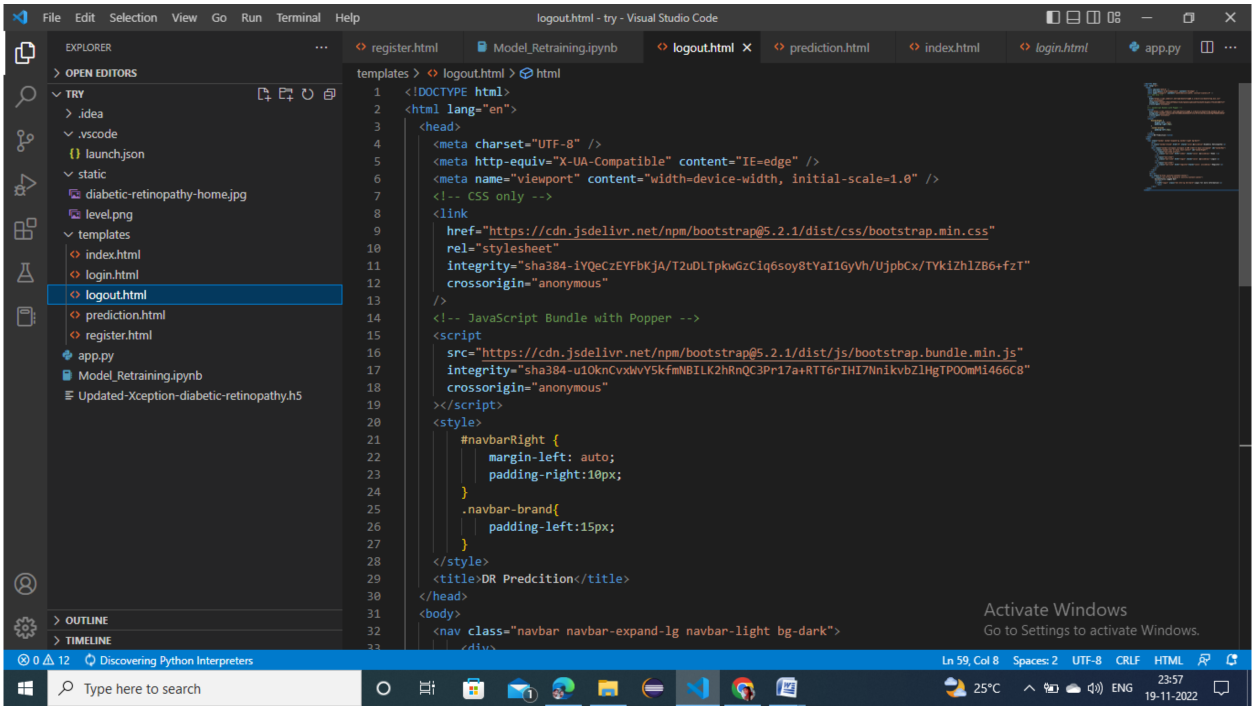
- Epochs: an integer and number of epochs we want to train our model for.

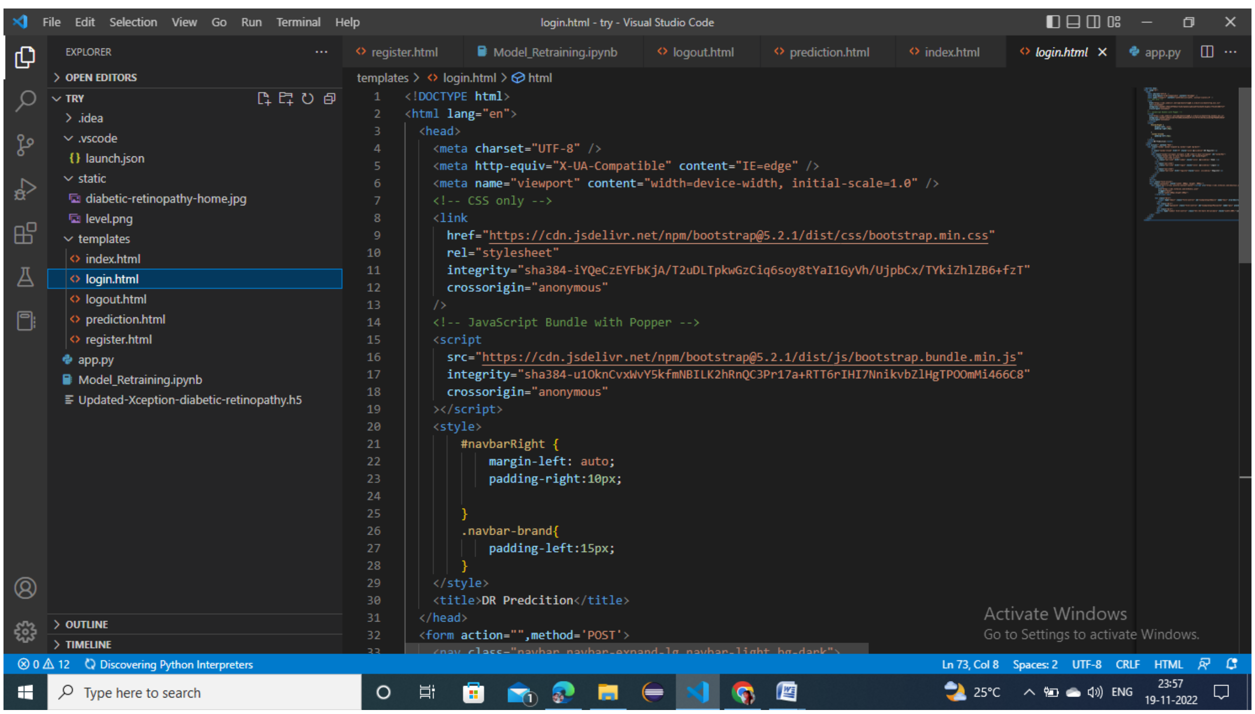
- validation_data can be either:
  - an inputs and targets list
  - a generator
  - an inputs, targets, and sample_weights list which can be used to evaluate the loss and metrics for any model after any epoch has ended.
- validation_steps: only if the validation_data is a generator then only this argument can be used. It specifies the total number of steps taken from the generator before it is stopped at every epoch and its value is calculated as the total number of validation data points in your dataset divided by the validation batch size.

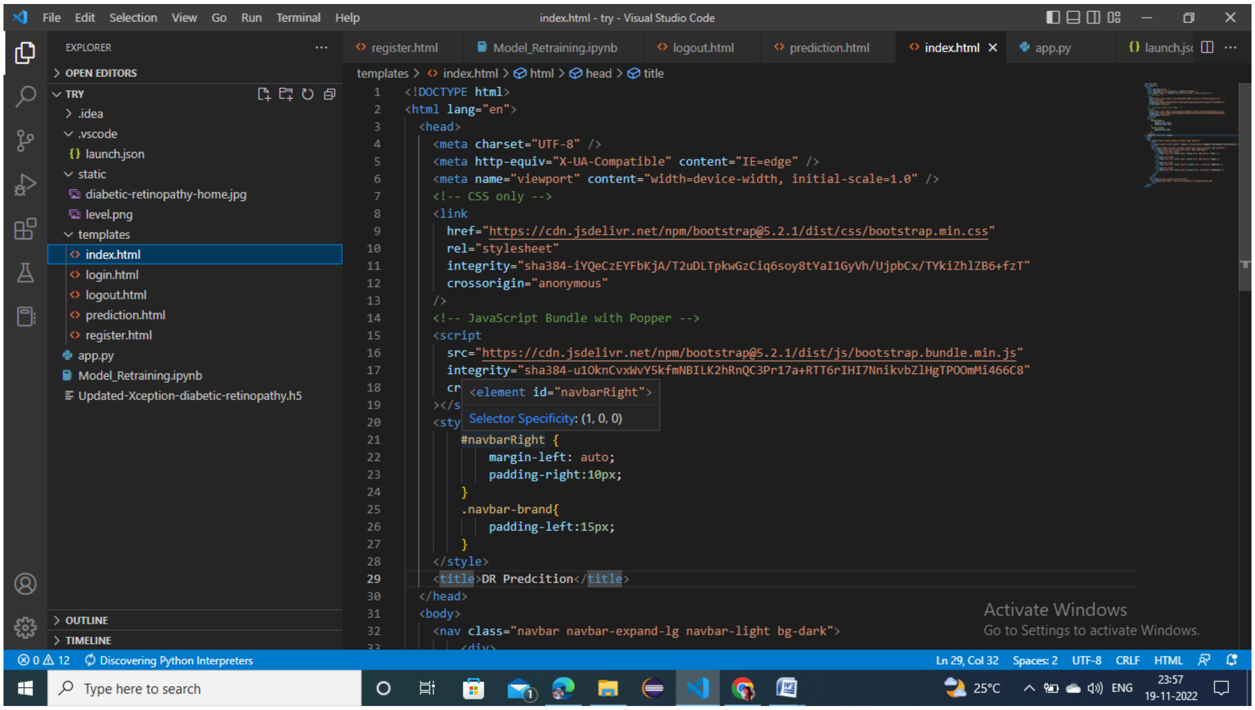



app.py - try - Visual Studio Code
EXPLORER
OPEN EDITORS
TRY
.idea
.vscode
launch.json
static
diabetic-retinopathy-home.jpg
level.png
templates
index.html
login.html
logout.html
prediction.html
register.html
app.py
Model_Retraining.ipynb
Updated-Xception-diabetic-retinopathy.h5
import numpy as np
import os
from tensorflow import keras
from keras import models
from keras.models import load_model
from keras.preprocessing import image
from keras.applications.inception_v3 import preprocess_input
import requests
from flask import Flask, request, render_template, redirect, url_for
from cloudant.client import Cloudant
model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
app = Flask(__name__)
# Authenticate using an IAM API key
client = Cloudant.iam('367e91e7-6150-4f63-92f4-24625af53457-bluemix',
'EqKm5BOKxzGLIm9YsFnXKJ66ywOyL9tDVK9oN0_FPD4G', connect=True)
# Create a database using an initialized client
my_database = client.create_database('my_db')
if my_database.exists():
print("Database '{0}' successfully created.".format('my_db'))
# default home page or route
@app.route('/')
def index():
return render_template('index.html')
@ app.route('/index')
def home():
return render_template("index.html")
'''@ app.route('/register')
def register():
return render_template("register.html")'''
Activate Windows
Go to Settings to activate Windows.
Ln 17, Col 48 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit
Type here to search
25°C ENG 23:57 19-11-2022


prediction.html - try - Visual Studio Code
templates > prediction.html > html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="ano
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
<style>
#navbarRight {
margin-left: auto;
padding-right: 10px;
}
.navbar-brand {
padding-left: 15px;
}
.row {
width: 90%;
}
</style>
<title>DR Predcition</title>
</head>
<body>
Activate Windows
Go to Settings to activate Windows.
Ln 63, Col 8 Spaces: 2 UTF-8 CRLF HTML
Type here to search
25°C ENG 23:57 19-11-2022


logout.html - try - Visual Studio Code
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- CSS only -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
/>
<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
></script>
<style>
#navbarRight {
margin-left: auto;
padding-right:10px;
}
.navbar-brand{
padding-left:15px;
}
</style>
<title>DR Predcition</title>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-dark">


login.html - try - Visual Studio Code
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- CSS only -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous"
/>
<!-- JavaScript Bundle with Popper -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"
></script>
<style>
#navbarRight {
margin-left: auto;
padding-right:10px;
}
.navbar-brand{
padding-left:15px;
}
</style>
<title>DR Predcition</title>
</head>
<form action="",method='POST'>



8. TESTING :

8.1: AcceptanceTesting UAT Execution & Report Submission

- PurposeofDocument

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

- DefectAnalysis

  This report shows the number of resolved or closed bugs a teach severity level,and howthey were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
ByDesign	1	0	0	0	1
Duplicate	4	1	3	0	8
External	1	3	0	0	4
Fixed	2	4	4	2	12

NotReproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won'tFix	0	0	0	0	0
Totals	8	8	4	2	22

- TestCaseAnalysis

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

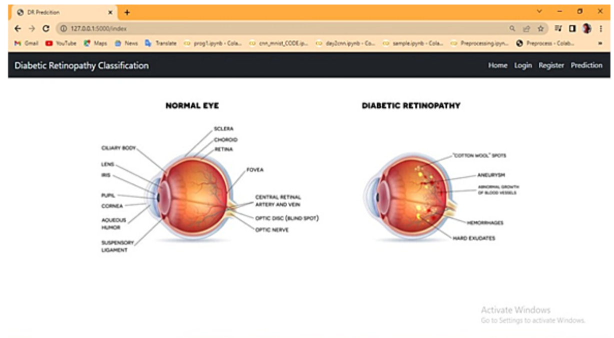
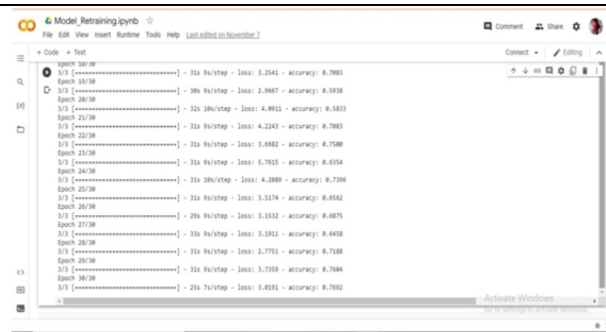
Section	TotalCases	NotTested	Fail	Pass
PrintEngine	5	0	0	5
ClientApplication	10	0	0	10
Security	2	0	0	2
OutsourceShipping	0	0	0	0
ExceptionReporting	2	0	0	2
FinalReportOutput	4	0	0	4
VersionControl	2	0	0	2

9. RESULTS:

9.1: Performance Metrics:

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

S.No	Parameter	Values	Screenshot
1.	Model Summary	-	

			
2.	Accuracy	Training Accuracy – 76.92 Validation Accuracy -71.88	
3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	

10. CONCLUSION:

The huge population of diabetic patients and the prevalence of Diabetic Retinopathy among them have fostered a great demand in automatic DRdiagnosing systems. So far, a lot of achievements have been made and satisfactoryresults have been achieved in many sub problems like vessel segmentation, lesiondetection. However, these results are obtained on datasets relatively small and aresteps awayfromrealworldapplications.

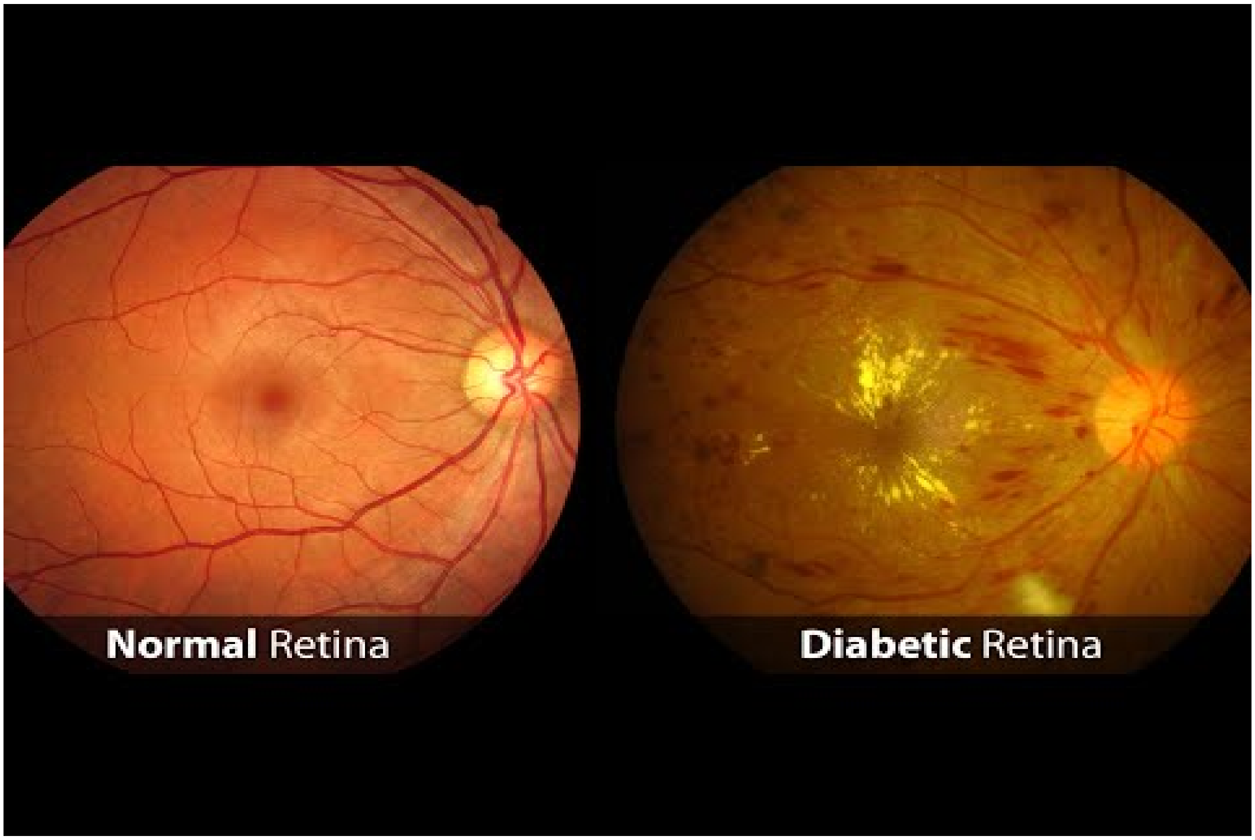
For clinical application, systems that can give DR severity directly are morefavourable and practical. However, current results for multi-class severity gradingare still not good enough for clinical application. In this work, we investigated theautomatic grading of DR using deep neural networks. We proposed a novel datasetthat is moderate in size and annotated with a new labeling scheme that is more useful for clinical practice.We proposed a preprocessing pipeline to change fundus images into a uniform format. We used the Inception-V3 network and a proposedmodification of it as our diagnostic models and evaluated the performance of themwith several mainstream CNN models. The experimental results demonstrate theefficiency of the models in diagnosing DR. Visualization and analysis of thetrained models provide insights into how the models make diagnoses using givenfundus images and justify the diagnostic ability of the models from a differentviewpoint. For clinical applications, the trained models are deployed on a cloudcomputing platform and provide pilot diagnostic services to several hospitals viathe internet. The performance of the system in the clinical evaluation demonstratesthe efficiency of this work. In the future, data from more equipments will beincluded, and a broader pilot study will be launched. The accumulated data will be further used to improve the accuracy of the models.

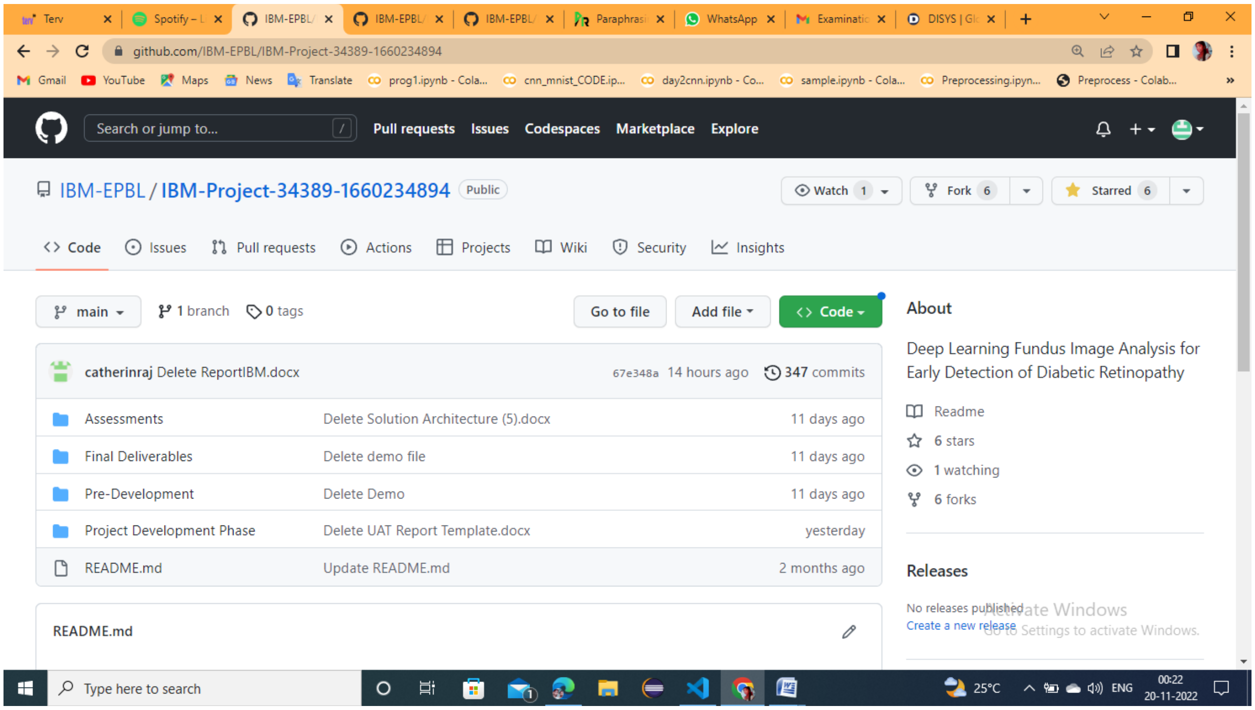
For clinical application, systems that can give DR severity directly are morefavourable and practical. However, current results for multi-class severity gradingare still not good enough for clinical application. In this work, we investigated theautomatic grading of DR using deep neural networks. We proposed a novel datasetthat is moderate in size and annotated with a new labeling scheme that is more useful for clinical practice.We proposed a preprocessing pipelinetochangefundusimages into a uniform format. We used the Inception-V3 network and a proposed modification of it as our diagnostic models and evaluated the performance of themwith several mainstream CNN models. The experimental results demonstrate theefficiency of the models in diagnosing DR. Visualization and analysis of thetrained models provide insights into how the models make diagnoses using givenfundus images and justify the diagnostic ability of the models from a different viewpoint. For clinical applications, the trained models are deployed on a cloudcomputing platform and provide pilot diagnostic services to several hospitals viathe internet. The performance of the system in the clinical evaluation demonstratesthe efficiency of this work. In the future, data from more equipments will beincluded, and a broader pilot study will be launched. The accumulated data will befurther usedtoimprovetheaccuracyofthemodels.

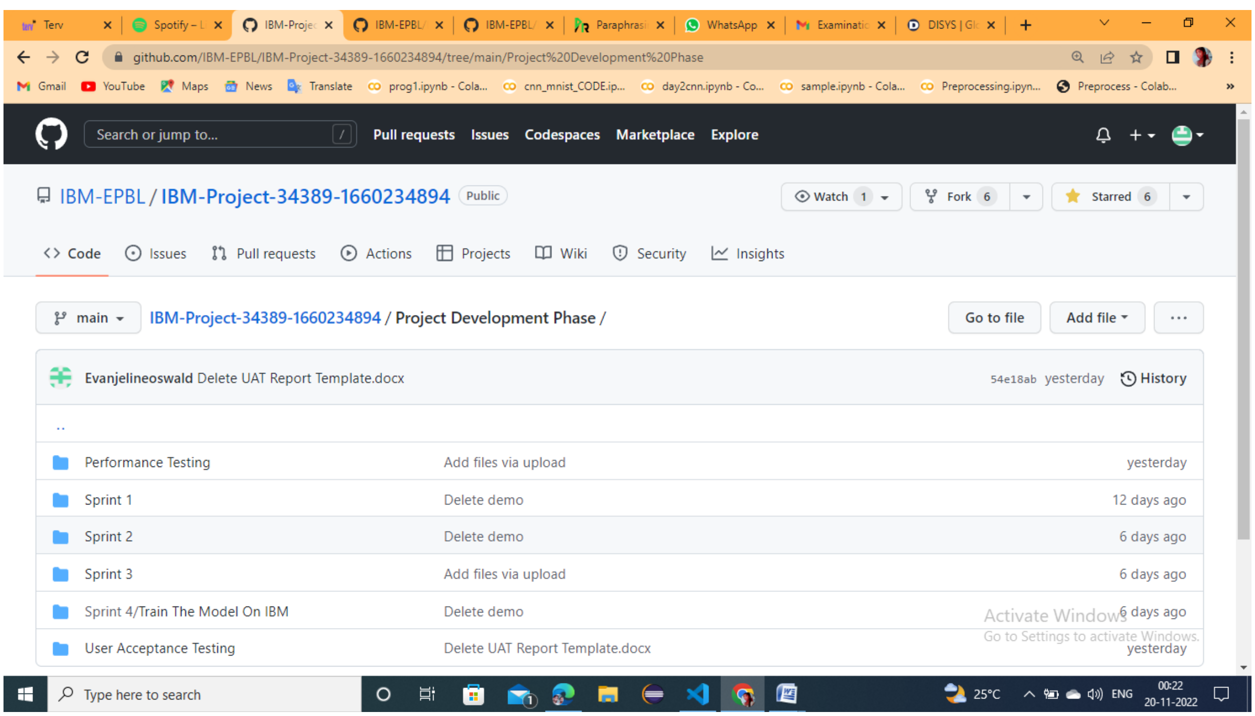
11.APPENDIX:

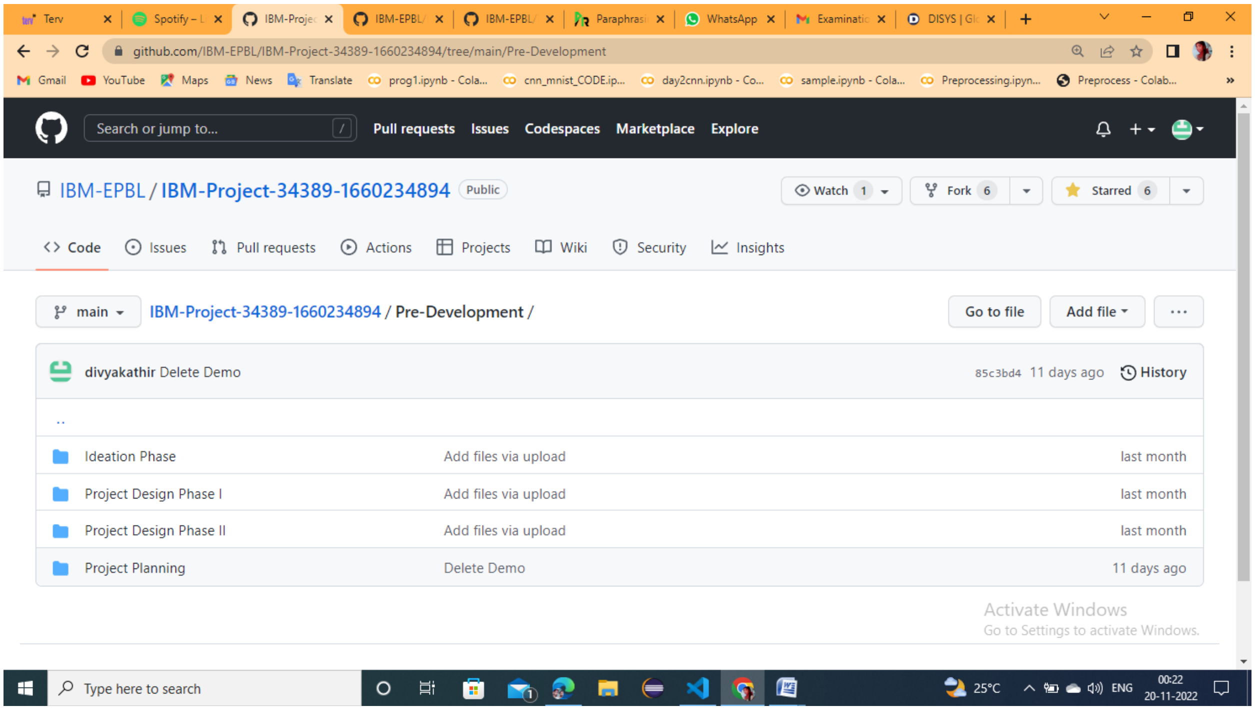
Github Link:
https://github.com/IBM-EPBL/IBM-Project-34389-1660234894

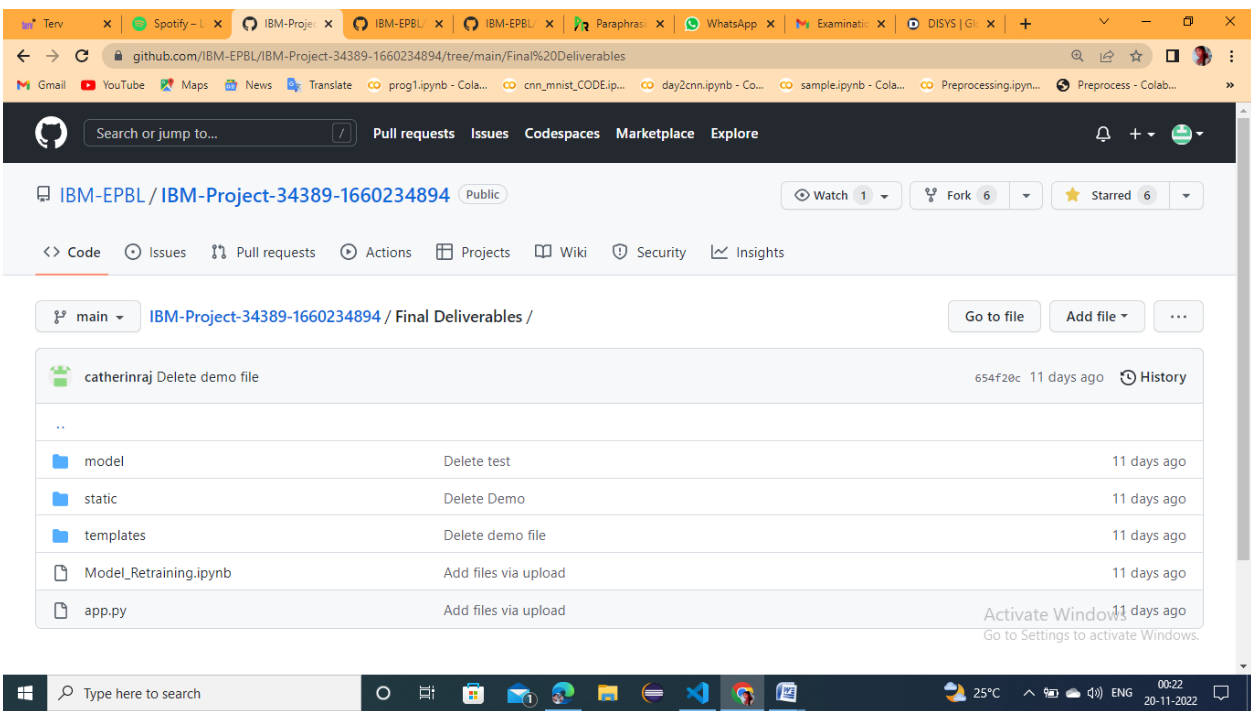
Demo Link:




github.com/IBM-EPBL/IBM-Project-34389-1660234894
Search or jump to...
Pull requests Issues Codespaces Marketplace Explore
IBM-EPBL / IBM-Project-34389-1660234894 Public
Watch 1
Fork 6
Starred 6
Code Issues Pull requests Actions Projects Wiki Security Insights
main
1 branch
0 tags
Go to file
Add file
Code
catherinraj Delete ReportIBM.docx
67e348a 14 hours ago
347 commits
Assessments Delete Solution Architecture (5).docx 11 days ago
Final Deliverables Delete demo file 11 days ago
Pre-Development Delete Demo 11 days ago
Project Development Phase Delete UAT Report Template.docx yesterday
README.md Update README.md 2 months ago
README.md
About
Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Readme
6 stars
1 watching
6 forks
Releases
No releases published
Create a new release
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C
ENG
00:22
20-11-2022


github.com/IBM-EPBL/IBM-Project-34389-1660234894/tree/main/Project%20Development%20Phase
Search or jump to...
Pull requests Issues Codespaces Marketplace Explore
IBM-EPBL / IBM-Project-34389-1660234894 Public
Watch 1
Fork 6
Starred 6
Code Issues Pull requests Actions Projects Wiki Security Insights
main
IBM-Project-34389-1660234894 / Project Development Phase /
Go to file
Add file
Evanjelineoswald Delete UAT Report Template.docx
54e18ab yesterday
History
Performance Testing Add files via upload yesterday
Sprint 1 Delete demo 12 days ago
Sprint 2 Delete demo 6 days ago
Sprint 3 Add files via upload 6 days ago
Sprint 4/Train The Model On IBM Delete demo 6 days ago
User Acceptance Testing Delete UAT Report Template.docx yesterday
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C
ENG
00:22
20-11-2022


github.com/IBM-EPBL/IBM-Project-34389-1660234894/tree/main/Pre-Development
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
IBM-EPBL / IBM-Project-34389-1660234894
Public
Watch 1
Fork 6
Starred 6
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
main
IBM-Project-34389-1660234894 / Pre-Development /
Go to file
Add file
divyakathir Delete Demo
85c3bd4 11 days ago
History
Ideation Phase
Add files via upload
last month
Project Design Phase I
Add files via upload
last month
Project Design Phase II
Add files via upload
last month
Project Planning
Delete Demo
11 days ago
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C
ENG
00:22
20-11-2022


github.com/IBM-EPBL/IBM-Project-34389-1660234894/tree/main/Final%20Deliverables
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
IBM-EPBL / IBM-Project-34389-1660234894
Public
Watch 1
Fork 6
Starred 6
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
main
IBM-Project-34389-1660234894 / Final Deliverables /
Go to file
Add file
catherinraj Delete demo file
654f20c 11 days ago
History
model
Delete test
11 days ago
static
Delete Demo
11 days ago
templates
Delete demo file
11 days ago
Model_Retraining.ipynb
Add files via upload
11 days ago
app.py
Add files via upload
11 days ago
Activate Windows
Go to Settings to activate Windows.
Type here to search
25°C
ENG
00:22
20-11-2022