



Project School Certificate

Title :Diabetic Retinopathy Detection (Web-Application)

Faculty Incharge : Dr .Devika Rubi

Session Duration : 5/09/2022- 15/12/2022

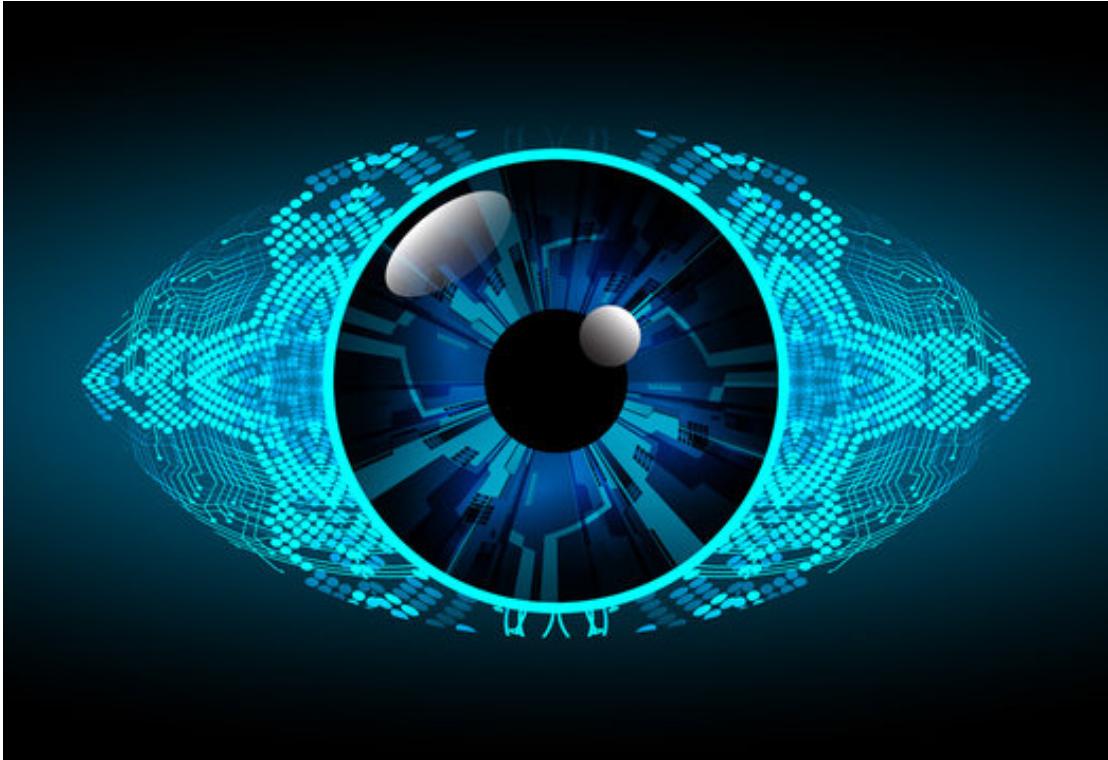
Name : D Nikhilesh

Roll Number : 21BD5A6602

Class : CSM - A

Signature of Faculty

Signature of student



Diabetic Retinopathy Detection

5/09/2022-15/12/2022

D Nikhilesh
21BD5A6602
III Year, CSM-A

Overview:

Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

Diabetic retinopathy affects up to 80 percent of those who have had both [type 1](#) and [type 2](#) diabetes for 20 years or more. In at least 90% of new cases, progression to more aggressive forms of sight threatening [retinopathy](#) could be reduced with proper treatment and monitoring of the eyes. The longer a person has diabetes, the higher his or her chances of developing diabetic retinopathy. Each year in the United States, diabetic retinopathy accounts for 12% of all new cases of blindness. It is also the leading cause of blindness in people aged 20 to 64.

So, We create a Web Application designed to classify the stages of diabetic retinopathy (DR) and predict the probability of its occurrence in the patient's retina.

Dataset Source:

The dataset used for the project is “Diabetic Retinopathy Detection” from Kaggle. Link to the dataset:

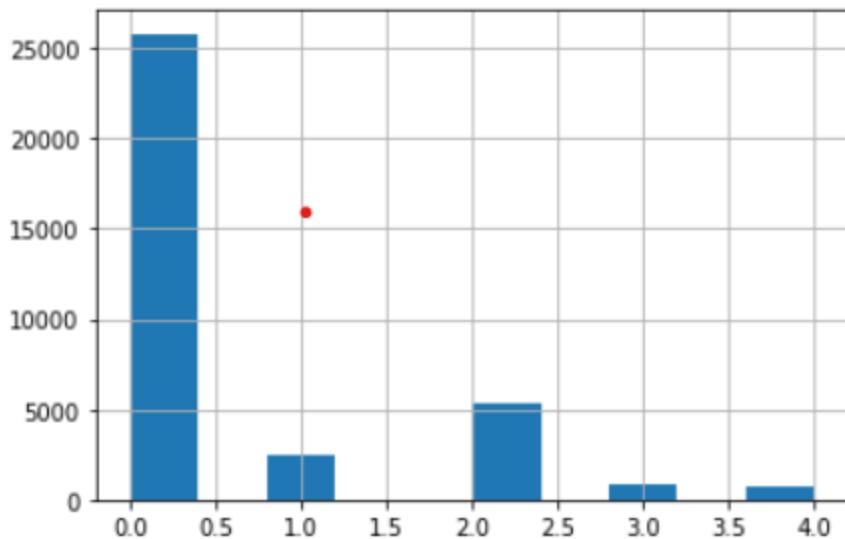
<https://www.kaggle.com/competitions/diabetic-retinopathy-detection/data>

The dataset consists of 35,122 Training images. A clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the following scale:

- 0-No DR
- 1-Mild DR
- 2-Moderate DR
- 3-Severe DR
- 4-Proliferate DR

The distribution of 5 classes is as follows.

```
0    25810  
2    5292  
1    2443  
3    873  
4    708  
Name: level, dtype: int64
```



Data Preprocessing:

Since, the data was very undistributed across various classes, the dataset size has been limited to 10,000 Training images. Class -3, Class-4 have been Augmented by 3 times, 4 times using `ImageDataGenerator(rotation_range=5, shear_range=0.1, zoom_range=0.1, horizontal_flip=True)` function from `keras.preprocessing.image` library,making class 3 a total of 2619 and class 4 a total of 2832 images. Finally, the dataset consists of 2000 images from each class. The images are then resized to 128*128 .

Model Used - MobileNet Deep Learning:

MobileNet uses **depth wise separable convolutions**. It significantly **reduces the number of parameters** when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depthwise separable convolution is made from two operations.

1. Depth Wise Convolution
2. Point Wise Convolution

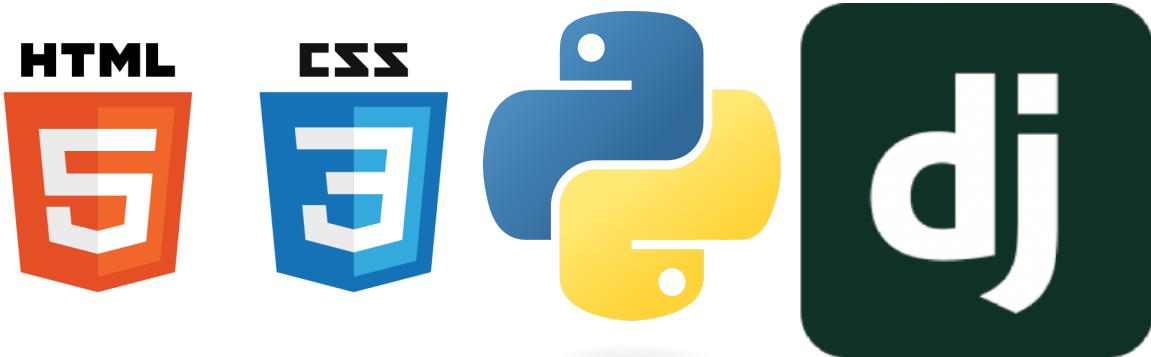
MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

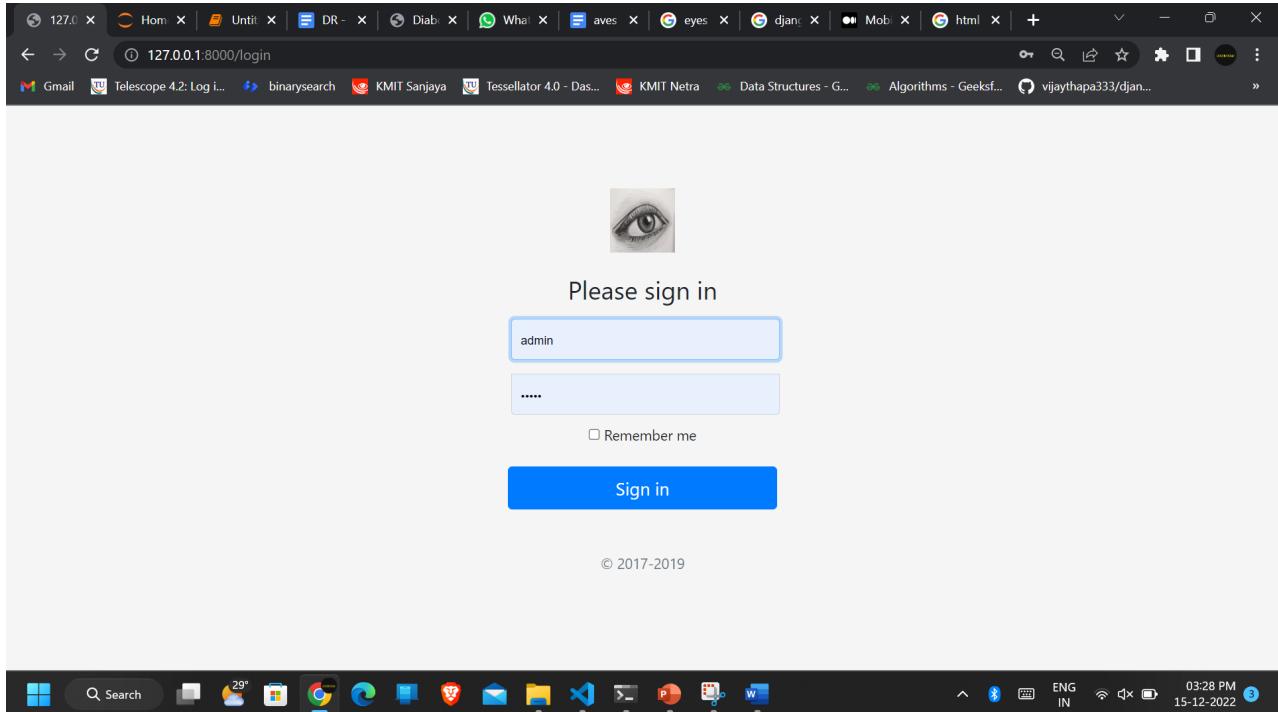
Tech Stack:

- Diabetic Retinopathy is detected using MobileNet CNN Algorithm.
- Front-End accepts the Patient Information , Eye images and displays the results, Front end is built using HTML & CSS
- Django is used as a database and Python serves as a backend framework.

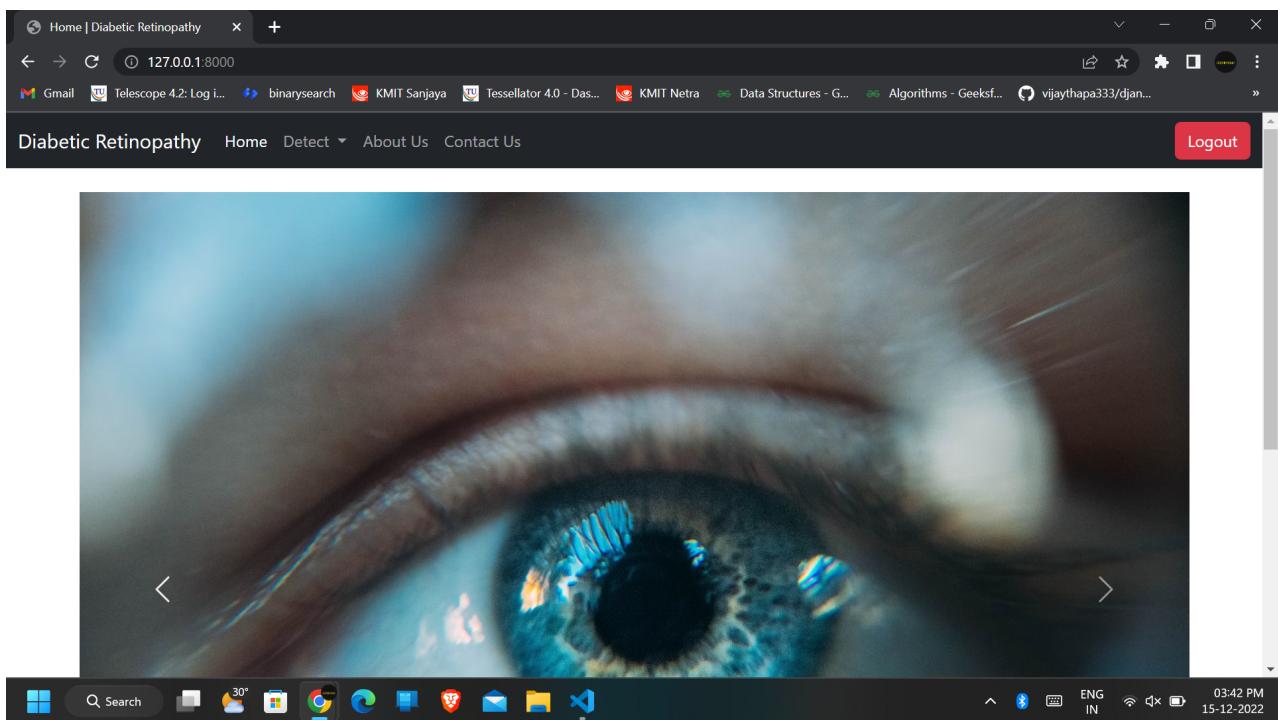


Project Screenshots:

1. Login Page



2. Home Page



3. Contact Page

A screenshot of a web browser window showing a contact form. The title bar says "Contact | Diabetic Retinopathy". The address bar shows "127.0.0.1:8000/contact". Below the address bar is a toolbar with various icons for Gmail, Telescope, binarysearch, KMIT Sanjaya, Tessellator, KMIT Netra, Data Structures, Algorithms, and a GitHub repository. The main content area contains a dark background image of clouds. The contact form includes fields for Name (placeholder "Enter your Name"), Email address (placeholder "Enter Your Email"), and a message area (placeholder "Tell me about what you want to contact me for..."). A blue "Submit" button is at the bottom left of the form.

4. About Page

A screenshot of a web browser window showing an "About" page. The title bar says "About | Diabetic Retinopathy". The address bar shows "127.0.0.1:8000/about". Below the address bar is a toolbar with various icons. The main content area shows a navigation menu with "Diabetic Retinopathy", "Home", "Detect", "About Us", and "Contact Us". On the right side, there is a red "Logout" button. Below the menu, a paragraph of text describes diabetic retinopathy as a leading cause of blindness in developed countries. Further down, another paragraph discusses the progression and symptoms of the disease.

Signs and Symptoms

Nearly all people with diabetes develop some degree of retina damage ("retinopathy") over several decades with the disease. For many, that damage can only be detected by a retinal exam, and has no noticeable effect on vision. Over time, progressive retinal damage may appear on a retinal exam, first with small bulges in retinal blood vessels called microaneurysms. Then larger abnormalities in retinal vessels: cotton wool spots, hemorrhages, lipid deposits called "hard exudates", intraretinal microvascular abnormalities, and abnormal-looking retinal veins. Eventually, many progress to a stage where new blood vessels grow throughout the retina. These new blood vessels often break and bleed. Minor bleeding can cause dark floating spots obstructing vision; major bleeding can completely block vision. Around half of people with diabetic retinopathy develop swelling of the macula, called macular edema, which can begin at any time. If the swelling occurs near the center of the macula, it can cause vision disruptions ranging from mildly blurred vision to severe loss of the center of an affected person's visual field. Left untreated, around 30% of those with such swelling experience vision disruption over the next 3–5 years. Macular edema is the most common cause of vision loss in people with diabetic retinopathy. The repeated processes of blood vessel growth, swelling, and scarring can eventually cause retinal detachment, which manifests as the sudden appearance of dark floating spots, flashes of light, or blurred vision.



5. Detect Page

Screenshot of the 'Detect' page for Diabetic Retinopathy. The URL is 127.0.0.1:8000/detect.

The page contains fields for Patient Name, Email address, Gender, Age, Phone Number, and two file upload fields for Right Image and Left Image. A 'Submit' button is at the bottom.

Patient Name: Enter your Name

Email address: Enter Your Email

Gender: Enter your Gender

Age: Enter your Age

Phone Number: Enter Your Phone Number

Right Image:- Choose File No file chosen

left Image:- Choose File No file chosen

Submit

6. Output Page

Screenshot of the 'Result' page for Diabetic Retinopathy. The URL is 127.0.0.1:8000/result.

The page displays patient information and two eye images. The patient information includes:

- Patient Name: Nikhil Dhotre
- Email: nikhil.dhotre@gmail.com
- Gender: male
- Age: 21
- Phone Number: 9440621220

The results section shows:

- Result: Right Image
- Image: A fundus photograph of a retina showing a small red spot (microaneurysm).
- Label: No_DR
- Result: Left Image
- Image: A fundus photograph of a retina showing a small red spot (microaneurysm).
- Label: No_DR



7. Results Page

S.No	Name	Email	Phone No.	Gender	Age	Right Image	Left Image	R Result	L Result
15	Nikhilesh Dhotre	nikhileshdhotre@gmail.com	9440621220	male	19	Link	Link	No_DR	No_DR
16	Bhanu Prasad	kommubhanuprasad2@email.com	06300035072	male	20	Link	Link	Mild	Mild
17	Nikhilesh Dhotre	nikhileshdhotre@gmail.com	9440621220	male	21	Link	Link	No_DR	No_DR

8. After Logout

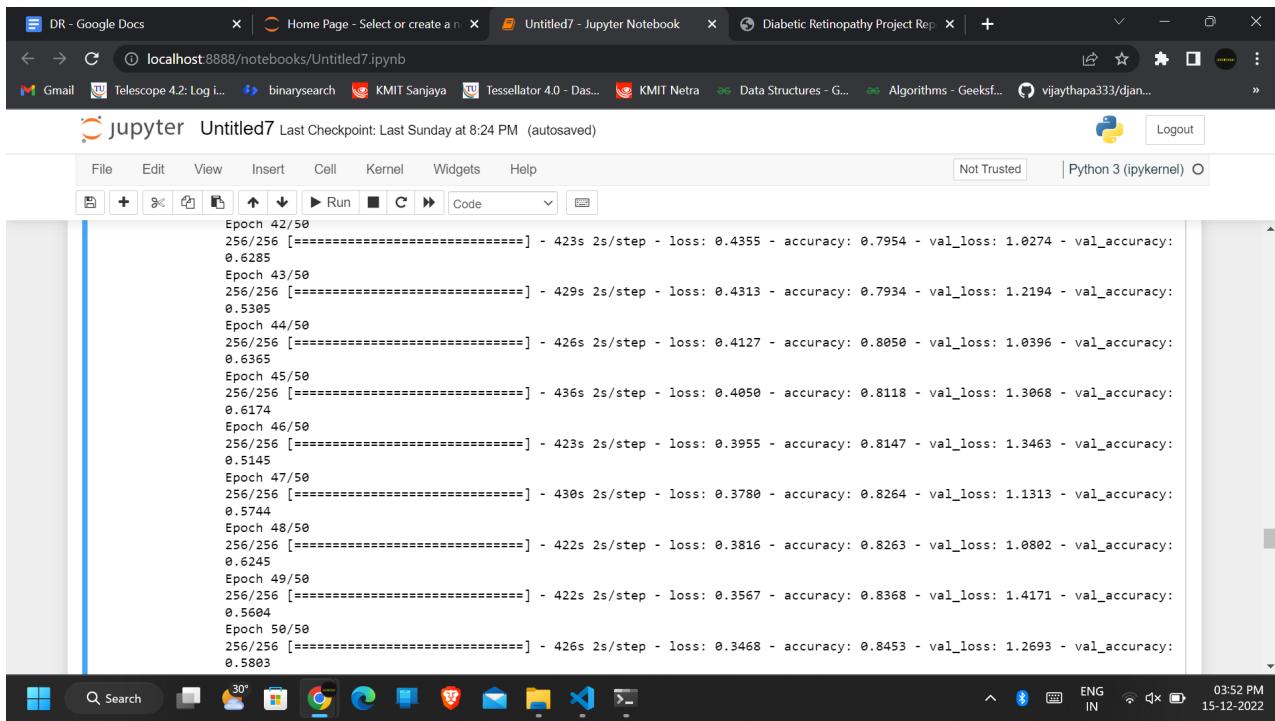
Please sign in

Remember me

Sign in

© 2017-2019

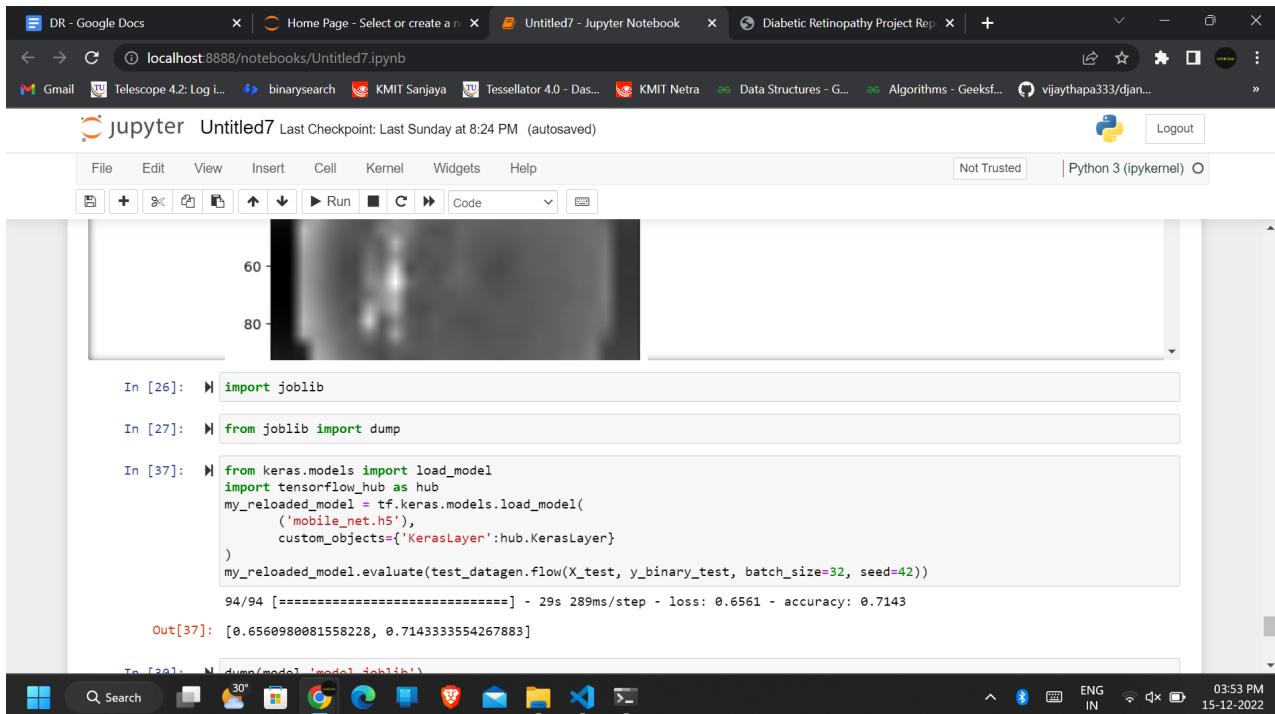
→ Train accuracy of Model:



A screenshot of a Jupyter Notebook interface. The title bar shows multiple tabs: DR - Google Docs, Home Page - Select or create a n..., Untitled7 - Jupyter Notebook, and Diabetic Retinopathy Project Rep... The notebook content displays a series of training logs from epoch 42 to 50. Each log entry shows a progress bar, step duration (e.g., 423s, 429s, 426s, 436s, 422s), loss values (e.g., 0.4355, 0.4313, 0.4050, 0.3955, 0.3816), and accuracy values (e.g., 0.7954, 0.7934, 0.8118, 0.8147, 0.8263). The logs also mention validation metrics like val_loss and val_accuracy.

```
Epoch 42/50
256/256 [=====] - 423s 2s/step - loss: 0.4355 - accuracy: 0.7954 - val_loss: 1.0274 - val_accuracy: 0.6285
Epoch 43/50
256/256 [=====] - 429s 2s/step - loss: 0.4313 - accuracy: 0.7934 - val_loss: 1.2194 - val_accuracy: 0.5385
Epoch 44/50
256/256 [=====] - 426s 2s/step - loss: 0.4127 - accuracy: 0.8050 - val_loss: 1.0396 - val_accuracy: 0.6365
Epoch 45/50
256/256 [=====] - 436s 2s/step - loss: 0.4050 - accuracy: 0.8118 - val_loss: 1.3068 - val_accuracy: 0.6174
Epoch 46/50
256/256 [=====] - 423s 2s/step - loss: 0.3955 - accuracy: 0.8147 - val_loss: 1.3463 - val_accuracy: 0.5145
Epoch 47/50
256/256 [=====] - 430s 2s/step - loss: 0.3780 - accuracy: 0.8264 - val_loss: 1.1313 - val_accuracy: 0.5744
Epoch 48/50
256/256 [=====] - 422s 2s/step - loss: 0.3816 - accuracy: 0.8263 - val_loss: 1.0802 - val_accuracy: 0.6245
Epoch 49/50
256/256 [=====] - 422s 2s/step - loss: 0.3567 - accuracy: 0.8368 - val_loss: 1.4171 - val_accuracy: 0.5604
Epoch 50/50
256/256 [=====] - 426s 2s/step - loss: 0.3468 - accuracy: 0.8453 - val_loss: 1.2693 - val_accuracy: 0.5883
```

→ Test Accuracy of Model:



A screenshot of a Jupyter Notebook interface. The title bar shows multiple tabs: DR - Google Docs, Home Page - Select or create a n..., Untitled7 - Jupyter Notebook, and Diabetic Retinopathy Project Rep... The notebook content shows a grayscale image of a retina fundus photograph. Below the image, several code cells are visible. Cell In [26] imports joblib. Cell In [27] imports dump from joblib. Cell In [37] imports load_model from keras.models and tensorflow_hub as hub, then reloads a model from 'mobile_net.h5' using custom_objects='KerasLayer'. It then evaluates the model on the test dataset, resulting in a loss of 0.6561 and an accuracy of 0.7143. The final output cell Out[37] shows the evaluated model's parameters: [0.6560980081558228, 0.714333554267883].

```
In [26]: import joblib
In [27]: from joblib import dump
In [37]: from keras.models import load_model
import tensorflow_hub as hub
my_reloaded_model = tf.keras.models.load_model(
    ('mobile_net.h5'),
    custom_objects={'KerasLayer':hub.KerasLayer})
my_reloaded_model.evaluate(test_datagen.flow(X_test, y_binary_test, batch_size=32, seed=42))
94/94 [=====] - 29s 289ms/step - loss: 0.6561 - accuracy: 0.7143
Out[37]: [0.6560980081558228, 0.714333554267883]
```