LuxPm Assignment

#Task 1 > Use the images and label it as apple

In [ ]:

```
```

In [ ]:

```
```

In [1]:

```
import os
```

In [ ]:

```
```

In [2]:

```
dataset_path=r".\dataset"
```

In [3]:

```
x_image= []
y_label= []
```

In [4]:

```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\rajat\anaconda3\l
ib\site-packages (4.5.3.56)
Requirement already satisfied: numpy>=1.17.3 in c:\users\rajat\anaconda3\l
ib\site-packages (from opencv-python) (1.19.5)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:

```
import cv2
```

In [6]:

```
labels = os.listdir(dataset_path)
```

In [7]:

```
labels
```

Out[7]:

```
['apple']
```

let's get each image and convert it into numeric array with similar size , also labeling it with label ="apple"

In [8]:

```
for directory in  labels:
    file_path=os.path.join(dataset_path, directory)
    for file in os.listdir(file_path):
        #print(file)
        full_image_path=os.path.join(file_path,file)
        print(full_image_path)
        image = cv2.imread(full_image_path)
        image = cv2.resize(image, (300,300))
        x_image.append(image.flatten())
        y_label.append(directory)
```

```
.\dataset\apple\apple_1.png
.\dataset\apple\apple_2.png
.\dataset\apple\apple_3.png
.\dataset\apple\apple_4.png
.\dataset\apple\apple_5.png
.\dataset\apple\apple_6.png
.\dataset\apple\apple_7.jpg
.\dataset\apple\apple_8.jpg
```

In [9]:

```
y_label
```

Out[9]:

```
['apple', 'apple', 'apple', 'apple', 'apple', 'apple', 'apple', 'apple']
```

let's convert it to numpy array

In [10]:

```
import numpy as np
```

In [11]:

```
x_image_np=np.array(x_image)
```

In [12]:

```
x_image_np.shape
```

Out[12]:

```
(8, 270000)
```

In [13]:

```
y_label_np=np.array(y_label)
```

In [14]:

```
y_label_np.shape
```

Out[14]:

```
(8,)
```

Task 1 accomplished

In [ ]:

In [ ]:

#Task 2 > Create a new directory named "test"

will create a new directory named "test" at the current path , only if it doesn't exist already

In [15]:

```
try:
    os.mkdir("test")
except:
    pass
```

Task 2 accomplished

In [ ]:

In [ ]:

In [ ]:

#Task 3> If I add a similar image into the directory, the machine should return "apple" as a value.

Building a Machine Learning classifer for our use case

In [16]:

```
from sklearn.model_selection import train_test_split
```

In [17]:

```
x_train,x_test, y_train, y_test= train_test_split(x_image_np, y_label_np, test_size=0.2,
```

In [18]:

```
x_train.shape
```

Out[18]:

```
(6, 270000)
```

In [19]:

```
x_test.shape
```

Out[19]:

```
(2, 270000)
```

In [20]:

```
from sklearn.ensemble import RandomForestClassifier
```

In [21]:

```
from sklearn.metrics import precision_score,recall_score,accuracy_score,f1_score,confusi
```

In [22]:

```
rf_model= RandomForestClassifier()
```

In [23]:

```
rf_model.fit(x_train,y_train)
```

Out[23]:

```
RandomForestClassifier()
```

In [24]:

```
y_pred=rf_model.predict(x_test)
```

In [25]:

```
accuracy_score(y_test,y_pred)
```

Out[25]:

```
1.0
```

In [26]:

```
precision_score(y_test,y_pred, average="weighted")
```

Out[26]:

```
1.0
```

In [27]:

```
y_pred
```

Out[27]:

```
array(['apple', 'apple'], dtype='<U5')
```

adding similar image to test directory to predict

In [ ]:

```

```

In [28]:

```
import urllib.request
```

In [29]:

```
urllib.request.urlretrieve("https://www.bostock.nz/wp-content/uploads/2019/09/organic-r
```

Out[29]:

```
('test/test_01.jpg', <http.client.HTTPMessage at 0x256c24efe50>)
```

let's import the image we just downloaded to test directory and convert to numpy array and predict

In [30]:

```
test_path=r".\test"
```

In [31]:

```
images_to_predict=os.listdir(test_path)
```

In [32]:

```
images_to_predict
```

Out[32]:

```
['test_01.jpg']
```

In [33]:

```
x =[]
```

In [34]:

```
        image = cv2.imread(images_to_predict[0])
        image = cv2.resize(image, (300,300))
        x.append(image.flatten())
```

In [35]:

```
x=np.array(x)
```

In [36]:

```
x.shape
```

Out[36]:

```
(1, 270000)
```

In [37]:

```
rf_model.predict(x)
```

Out[37]:

```
array(['apple'], dtype='<U5')
```

Hurray, our model predicted ("apple") which is correct class

Task 3 accomplished

In [ ]:

In [ ]:

In [ ]:

#Task4> Create an image file and provide us with steps for setting up the machine.

image file ->

i just used a random image of apple from google,and used it in our python code to predict

Steps ->

1> we can save and export the model using pickle module

2> we can use Flask or other python framework to deploy the model

i can also set custom file upload option in this ipython notebook

Note: this is just a simple basic model and it'll only give results like apple. as it was trained only on a single category. that's why we see accuracy as 1 as it's overfitting due to selection bias

but i hope with this little small efforts , i am able to accomplish the goals for the assignment

In [ ]:

Thank you

In [ ]: