



Course Name:	APPA	Semester:	V
Date of Performance:	01 / 10 / 2024	Batch No:	APPA 2
Faculty Name:	Prof. Deepa Jain	Roll No:	16014022096
Faculty Sign & Date:		Grade/Marks:	/25

Experiment No: 6

Title: Study Computer Vision Libraries of Python

Aim and Objective of the Experiment:

Implementation of following case study based on computervision using python programming.

- 1) Counting Coins
- 2) Face Detection
- 3) Optical Character Recognition

COs to be achieved:

CO1: Implementation of various Python Packages.

Theory

Face Detection:

In the Face Detection algorithm, it capture object of interest(face) in real time and to keep tracking of the same object. This is a simple example of how to detect face in Python. You can try to use training samples of any other object of your choice to be detected by training the classifier on required objects.

Here is the steps to download the requirements below.

Steps:

1. Download Python 2.7.x version, numpy and Opencv 2.7.x version. Check if your Windows either 32 bit or 64 bit is compatible and install accordingly.
2. Make sure that numpy is running in your python then try to install opencv.
3. Put the haarcascade_eye.xml & haarcascade_frontalface_default.xml files in the same folder(links given in below code).

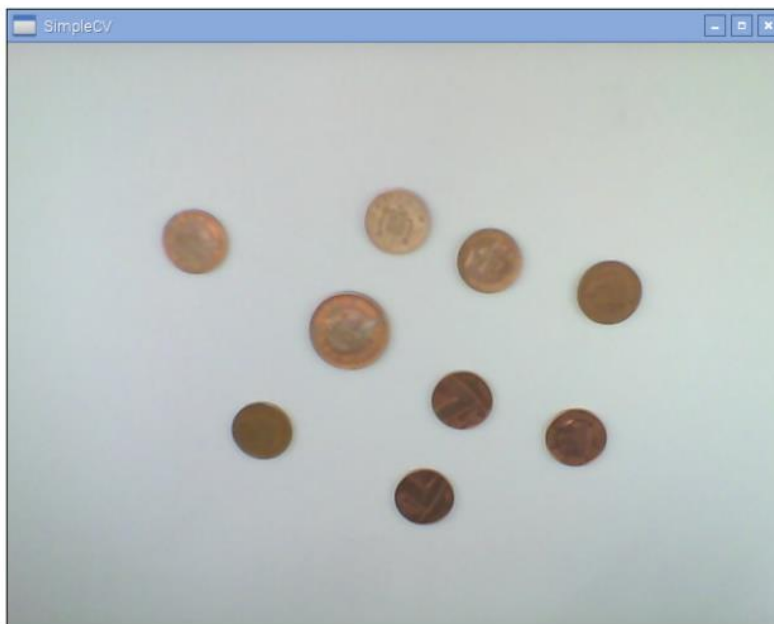
Optical Character Recognition

Python is widely used for analyzing the data but the data need not be in the required format always. In such cases, we convert that format (like PDF or JPG, etc.) to the text format, in order to analyze the data in a better way. Python offers many libraries to do this task.

Counting Coins:

Use SimpleCV and its findCircle function to provide a real-time count of the number of coins placed under

the webcam. This is one use of CV where you really need good lighting and a camera fixed in position. I used the setup shown in Figure 8-1. 8.4 Counting Coins | 183 Before writing a Python program that will simply tell you the number of coins your Raspberry Pi can see, you need to experiment with the SimpleCV console to get the parameters for circle recognition right.



Tools required:

Any python editor tool

Code:

- Write a Python Program for Face Detection, OCR and Counting Coins

FACE DETECTION:

```
import numpy as np
import cv2
import matplotlib.pyplot as plt

faceCascade = cv2.CascadeClassifier('EXPERIMENT
06/haarcascade_frontalface_default.xml')

image = cv2.imread('EXPERIMENT 06/demo.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray, scaleFactor=1.1,
minNeighbors=5,minSize=(30, 30))
print(f"Found {len(faces)} faces!")

image2 = image.copy()

for (x, y, w, h) in faces:
    cv2.rectangle(image2, (x, y), (x + w, y + h), (255, 0, 0), 2)

plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title('original image')

plt.subplot(1, 2, 2)
plt.imshow(image2)
plt.title('face cascade image')

plt.show()
cv2.destroyAllWindows()
```

OCR:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import pytesseract

pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'

print("demo hello")

image = cv2.imread('EXPERIMENT 06/demo2.jpg')
kernel = np.ones((2, 1), np.uint8)
img1 = cv2.erode(image, kernel, iterations=1)
img2 = cv2.dilate(image, kernel, iterations=1)
out_below = pytesseract.image_to_string(image)

print("OUTPUT: \n", out_below)

image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
img2_rgb = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)

plt.subplot(1, 2, 1)
plt.imshow(image_rgb)
plt.title('Original Image')

plt.subplot(1, 2, 2)
plt.imshow(img2_rgb)
plt.title('Dilated Image')

plt.show()
```

COUNTING COINS:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('EXPERIMENT 06/demo3.png')
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

blur = cv2.GaussianBlur(gray,(11,11),0)
canny = cv2.Canny(blur,30,150,3)
dilated=cv2.dilate(canny,(1,1),iterations=1)

cnt,hierarchy=cv2.findContours(dilated.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
image2 = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)

cv2.drawContours(image2,cnt,-1,(0,255,0),2)
print("Coins in the image are:", len(cnt))

plt.subplot(1, 2, 1)
plt.imshow(image)
plt.title('original image')

plt.subplot(1, 2, 2)
plt.imshow(image2)
plt.title('contour image of coins')

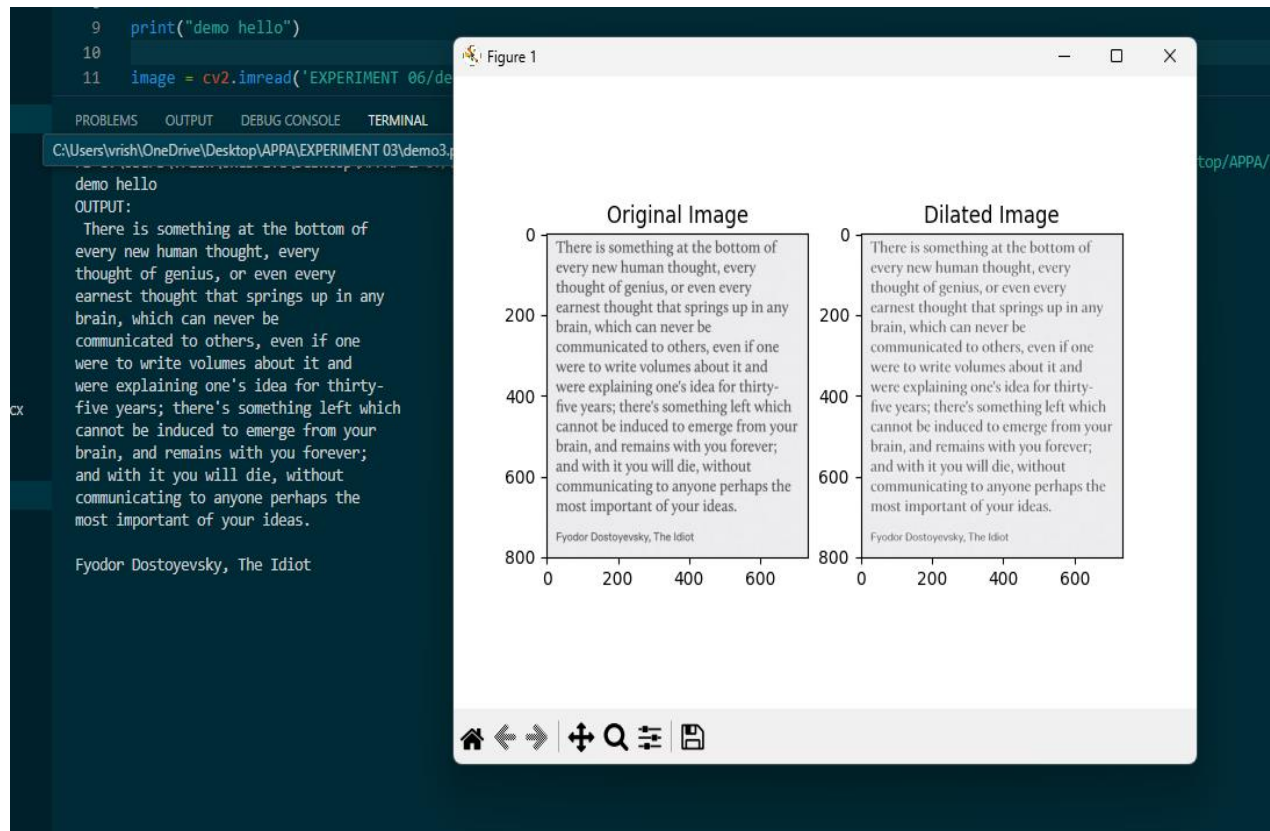
plt.show()
```

Output:

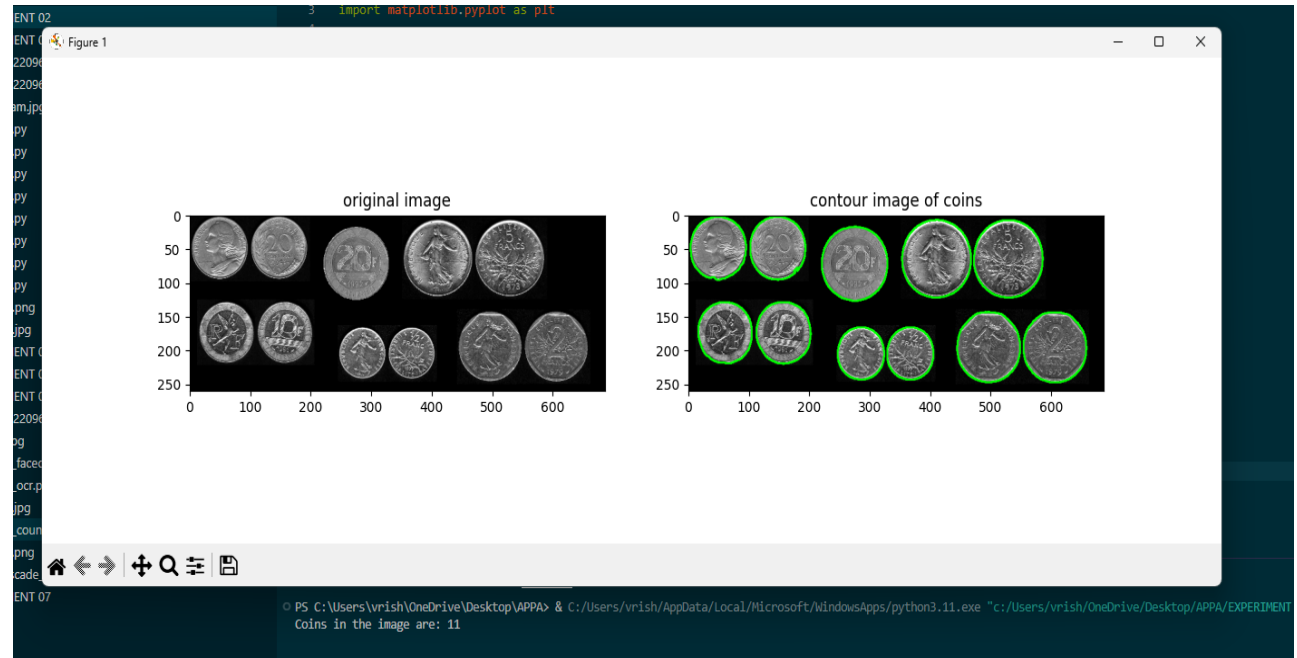
FACE DETECTION:



OCR:



COUNTING COINS:



Post Lab Subjective/Objective type Questions:

Write a python program for Face detection in Video capture

CODE:

```
import cv2

face_cascade = cv2.CascadeClassifier('EXPERIMENT
06/haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture('EXPERIMENT 06/demo4.mp4')

while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

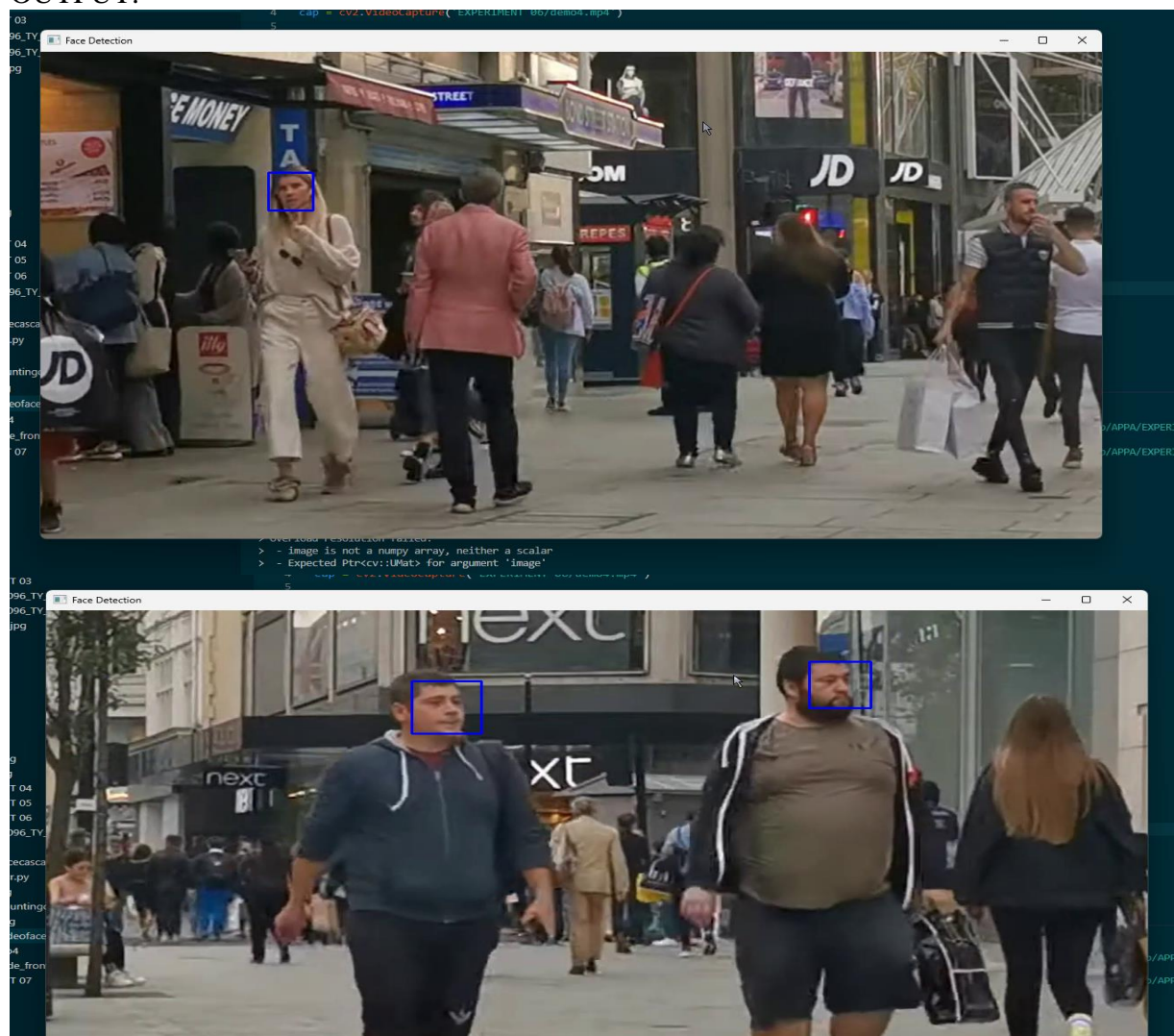
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

    cv2.imshow('Face Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```


OUTPUT:



Conclusion:

We have successfully explored Python's open-cv libraries to implement coin counting, face detection, and optical character recognition which helped in providing practical insights into leveraging tools like OpenCV and Tesseract for various image processing tasks.

Signature of faculty in-charge with Date: