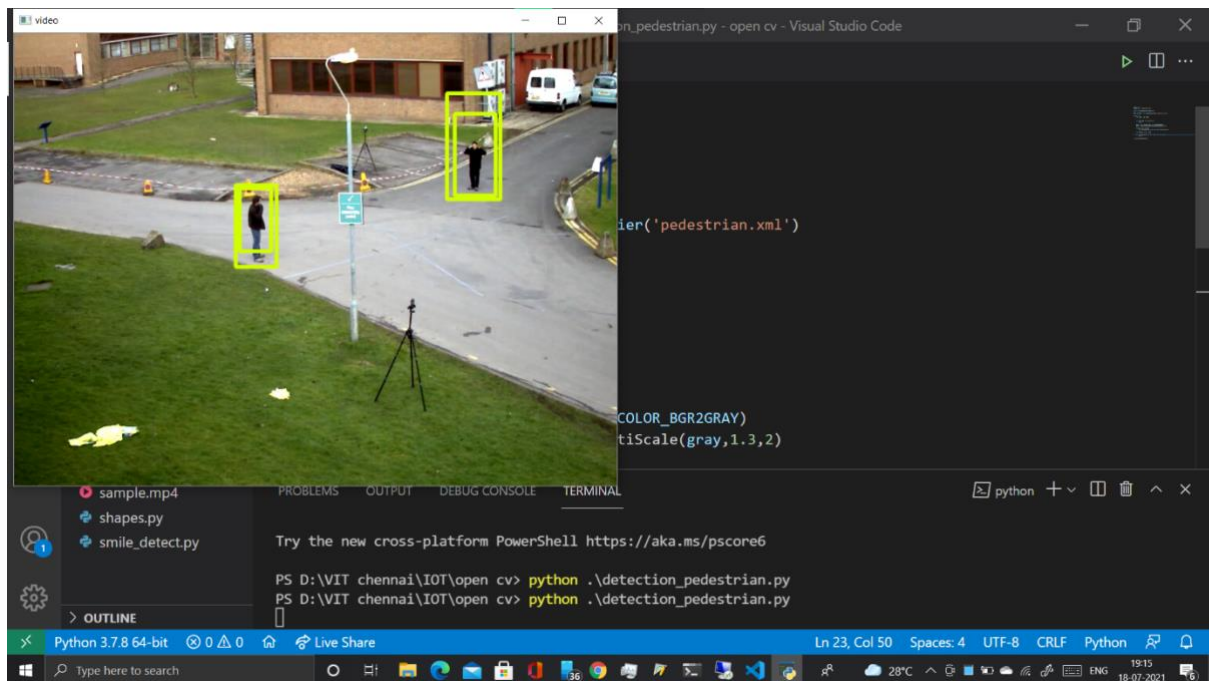
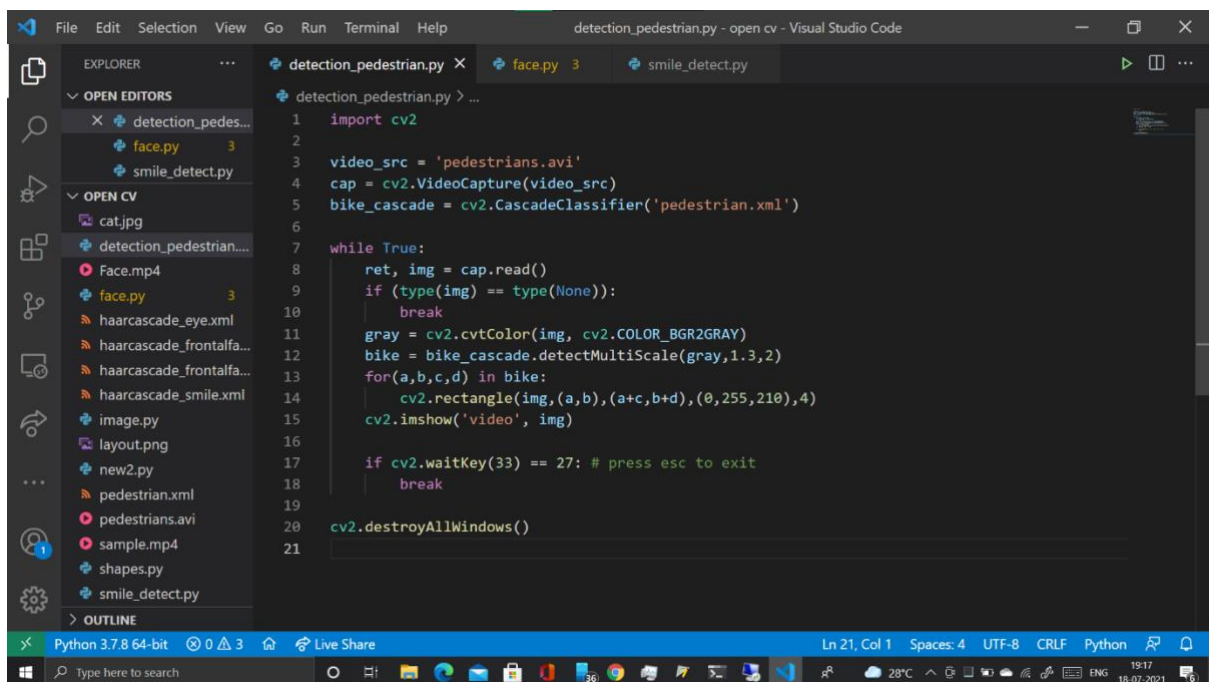


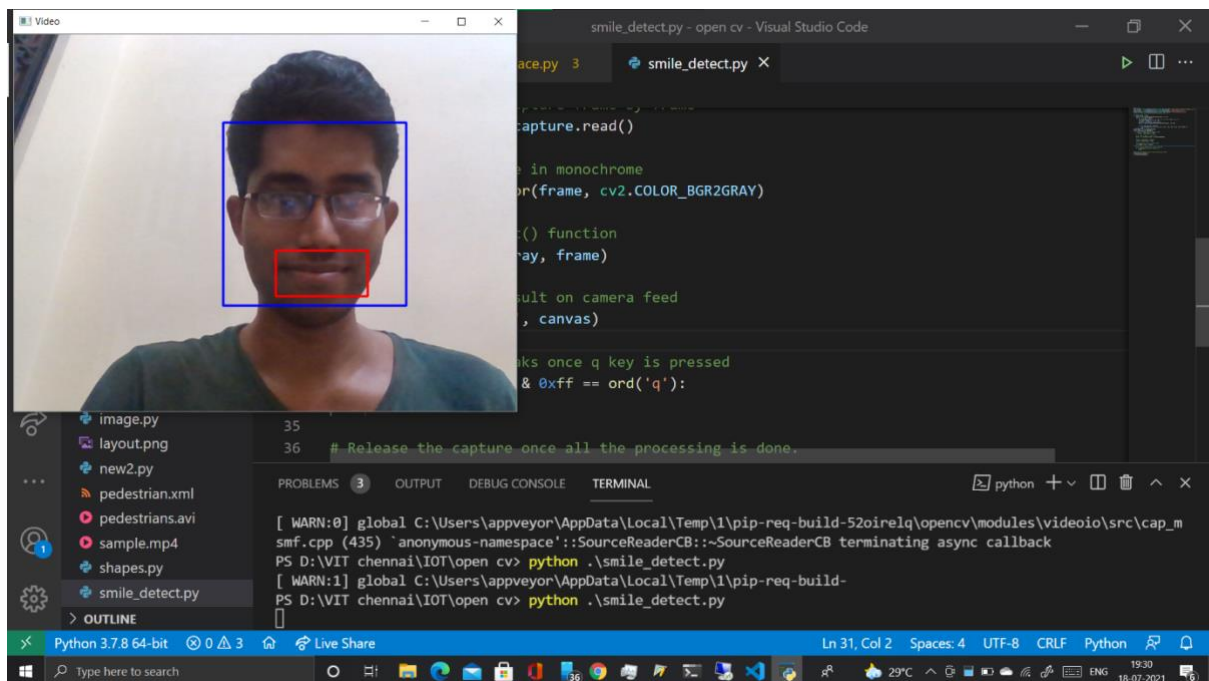
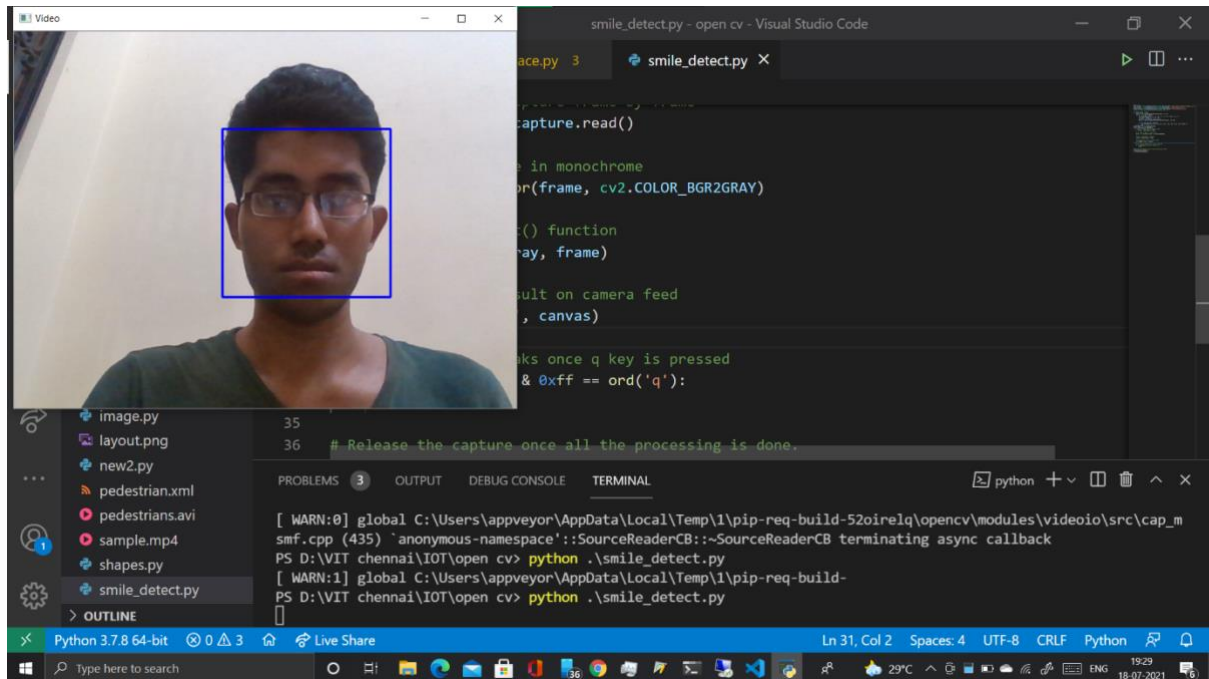
Pedestrian detection:



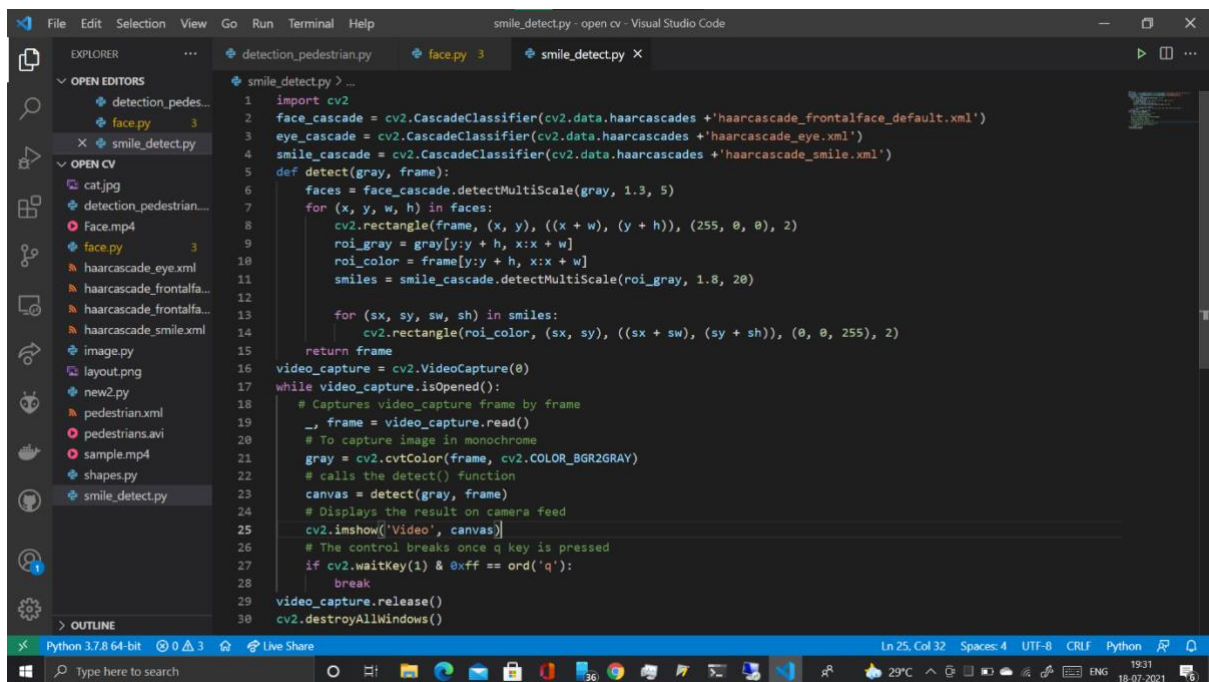
Python code:



Smile detection:



Python code:



```
1 import cv2
2 face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
3 eye_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_eye.xml')
4 smile_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_smile.xml')
5 def detect(gray, frame):
6     faces = face_cascade.detectMultiScale(gray, 1.3, 5)
7     for (x, y, w, h) in faces:
8         cv2.rectangle(frame, (x, y), ((x + w), (y + h)), (255, 0, 0), 2)
9         roi_gray = gray[y:y + h, x:x + w]
10        roi_color = frame[y:y + h, x:x + w]
11        smiles = smile_cascade.detectMultiScale(roi_gray, 1.8, 20)
12        for (sx, sy, sw, sh) in smiles:
13            cv2.rectangle(roi_color, (sx, sy), ((sx + sw), (sy + sh)), (0, 0, 255), 2)
14    return frame
15
16 video_capture = cv2.VideoCapture(0)
17 while video_capture.isOpened():
18     # Captures video_capture frame by frame
19     _, frame = video_capture.read()
20     # To capture image in monochrome
21     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
22     # calls the detect() function
23     canvas = detect(gray, frame)
24     # Displays the result on camera feed
25     cv2.imshow('Video', canvas)
26     # The control breaks once q key is pressed
27     if cv2.waitKey(1) & 0xFF == ord('q'):
28         break
29 video_capture.release()
30 cv2.destroyAllWindows()
```

Haar cascade classifier is uploaded in code section of Assignment6_code