Experiment-11

Centroid Tracking In a Given Video

Name : N U Praneeth Reddy Reg.No:21BAI1500

Aim: To perform centroid based object tracking on a given video.

Resources Used: Anaconda Python Environment
VSCode

Theory :

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Tasks:

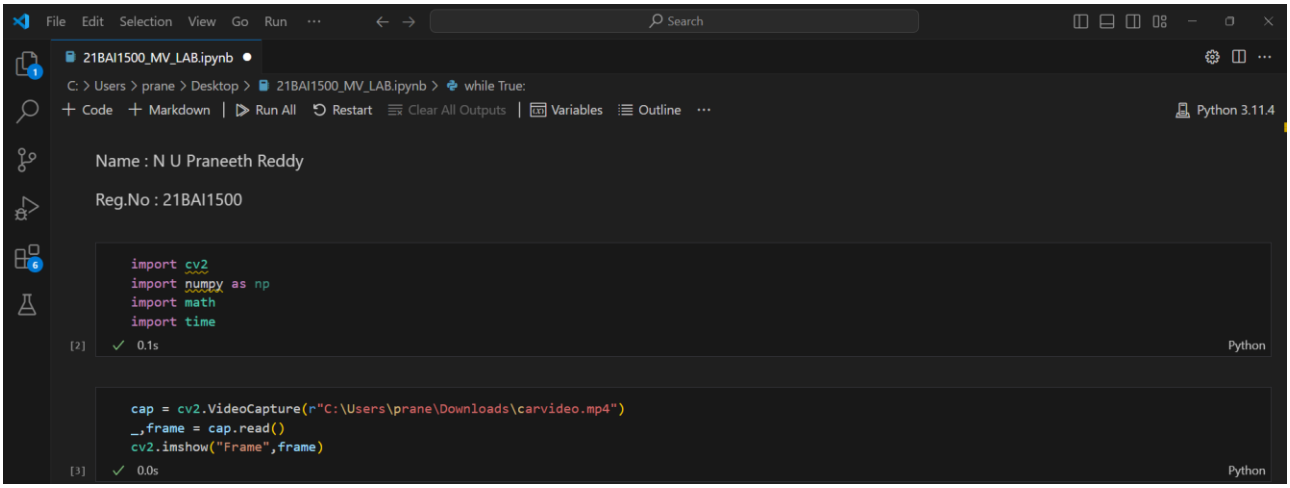
Import a sample video and perform the following steps for object tracking

a) Import video using cv2. Videocapture
b) Perform Background Subtraction
c) Detect the objects and Find contours
d) Determine the centroid and assign unique object ids
e) Finally track the objects in the video.

Procedure:

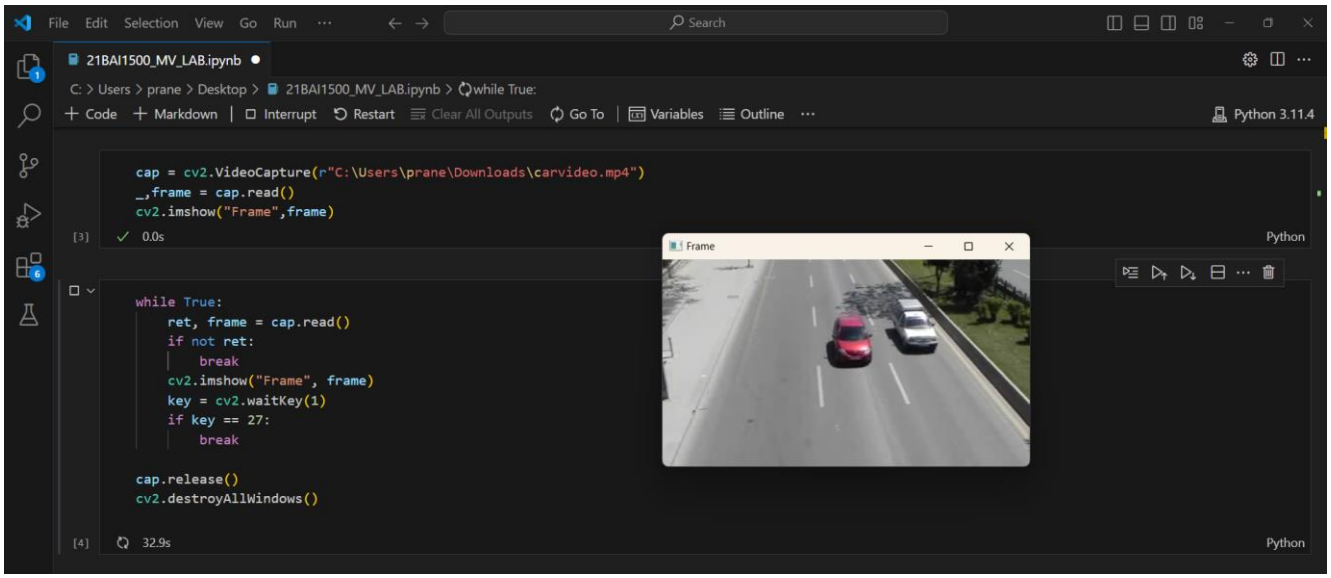
1)Open VSCode and create a new Jupyter Notebook.

2)Import the required libraries which are OpenCV and numpy along with math and time.

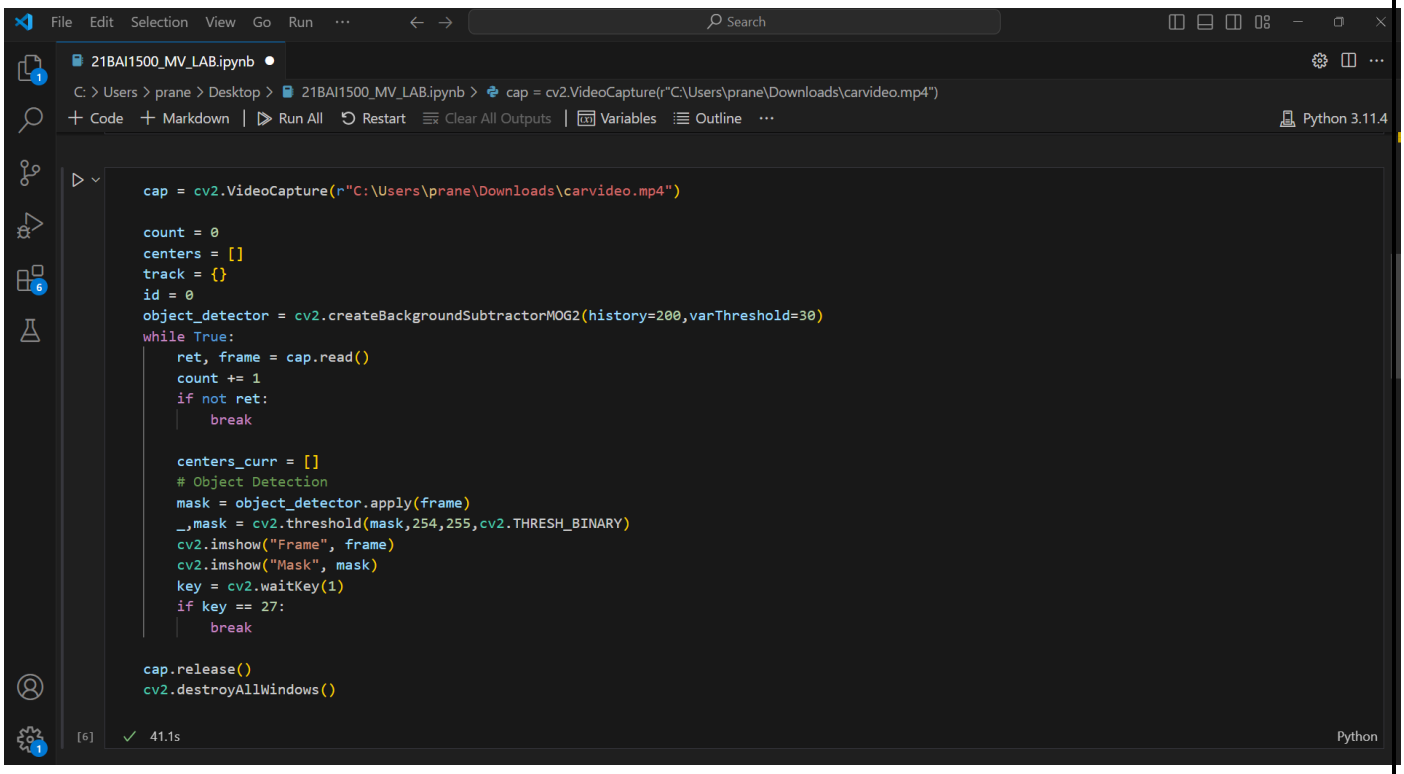


Task 1: Import video using cv2. Videocapture.

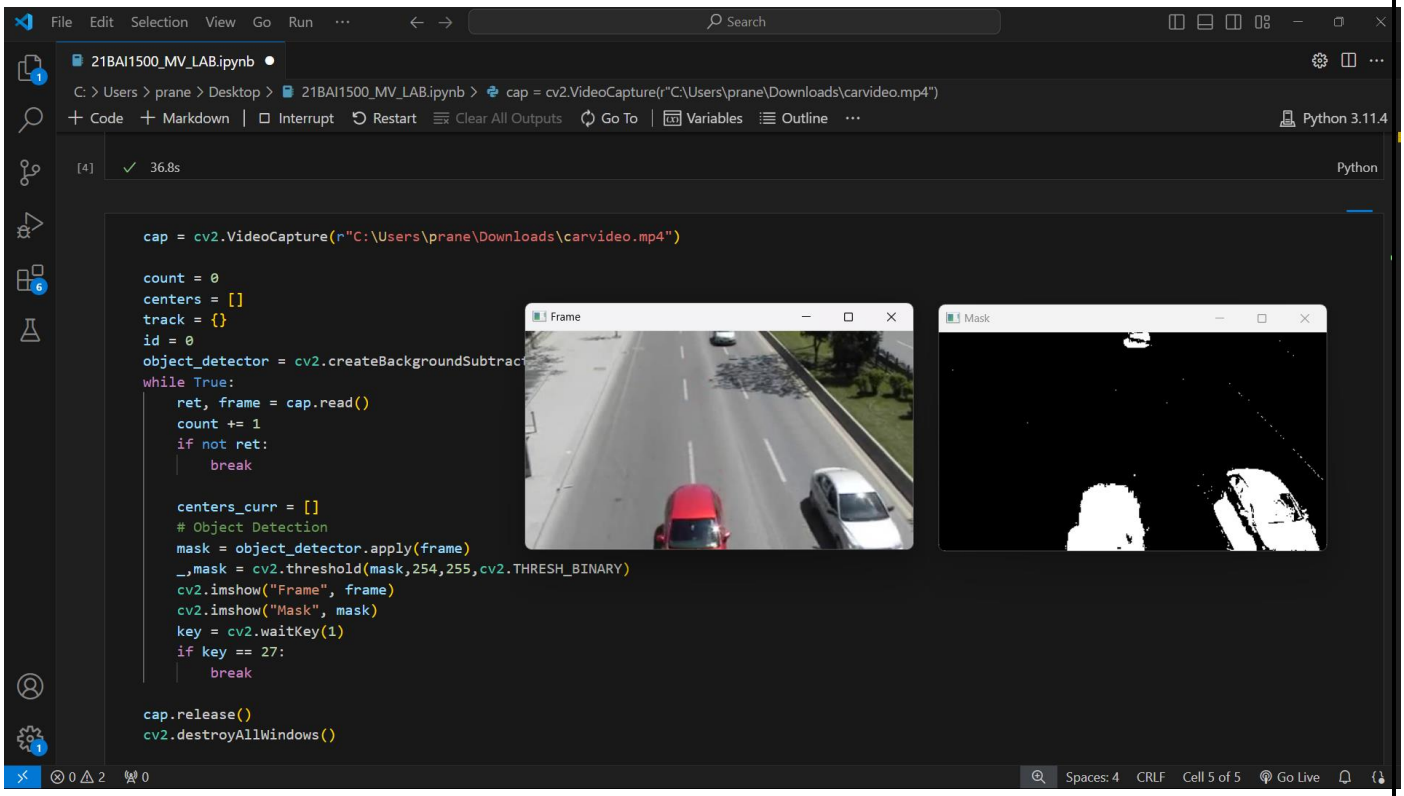
Read the given video using VideoCapture of the opencv library.



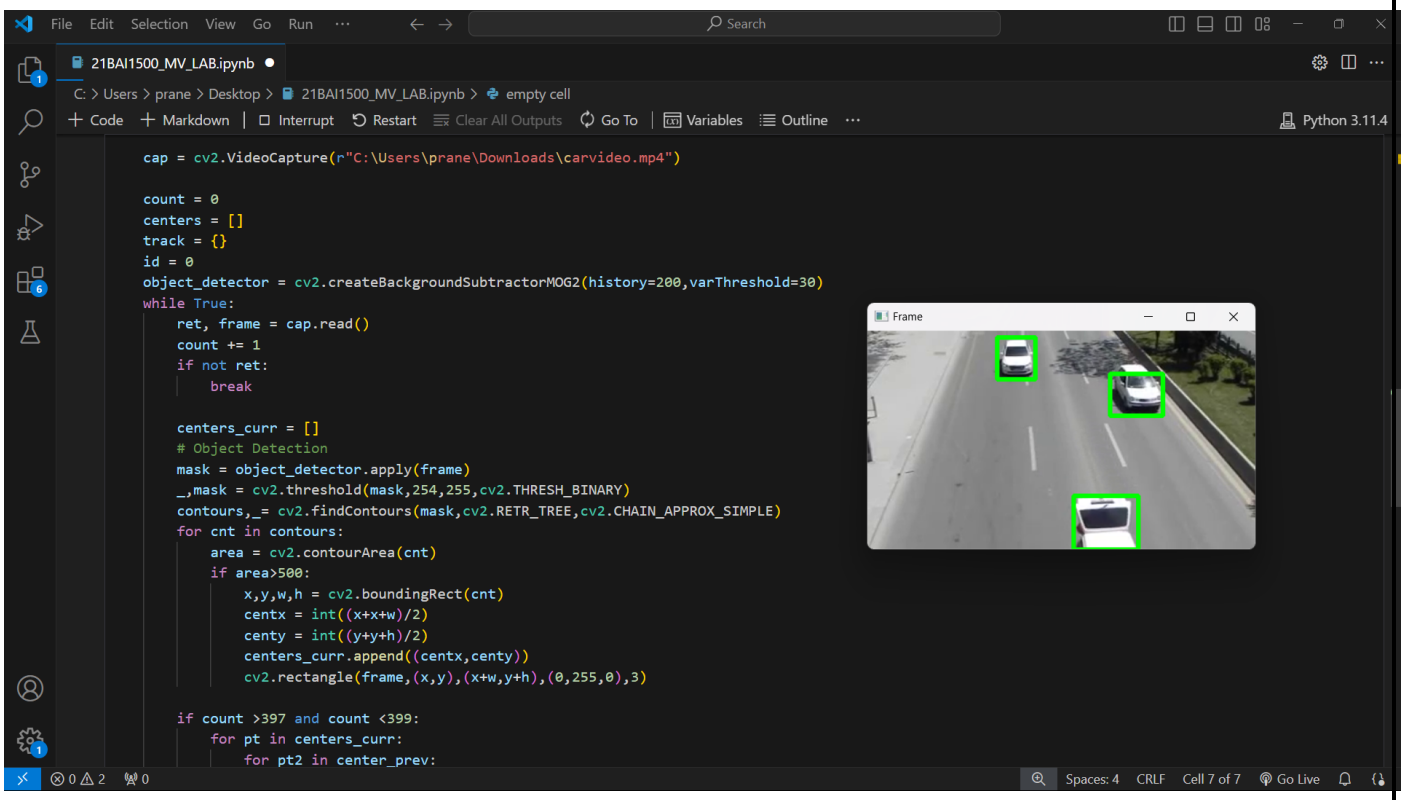
Task -2: Perform Background Subtraction



```
cap = cv2.VideoCapture(r"C:\Users\prane\Downloads\carvideo.mp4")

count = 0
centers = []
track = {}
id = 0
object_detector = cv2.createBackgroundSubtractorMOG2(history=200,varThreshold=30)
while True:
    ret, frame = cap.read()
    count += 1
    if not ret:
        break

    centers_curr = []
    # Object Detection
    mask = object_detector.apply(frame)
    _,mask = cv2.threshold(mask,254,255,cv2.THRESH_BINARY)
    cv2.imshow("Frame", frame)
    cv2.imshow("Mask", mask)
    key = cv2.waitKey(1)
    if key == 27:
        break

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



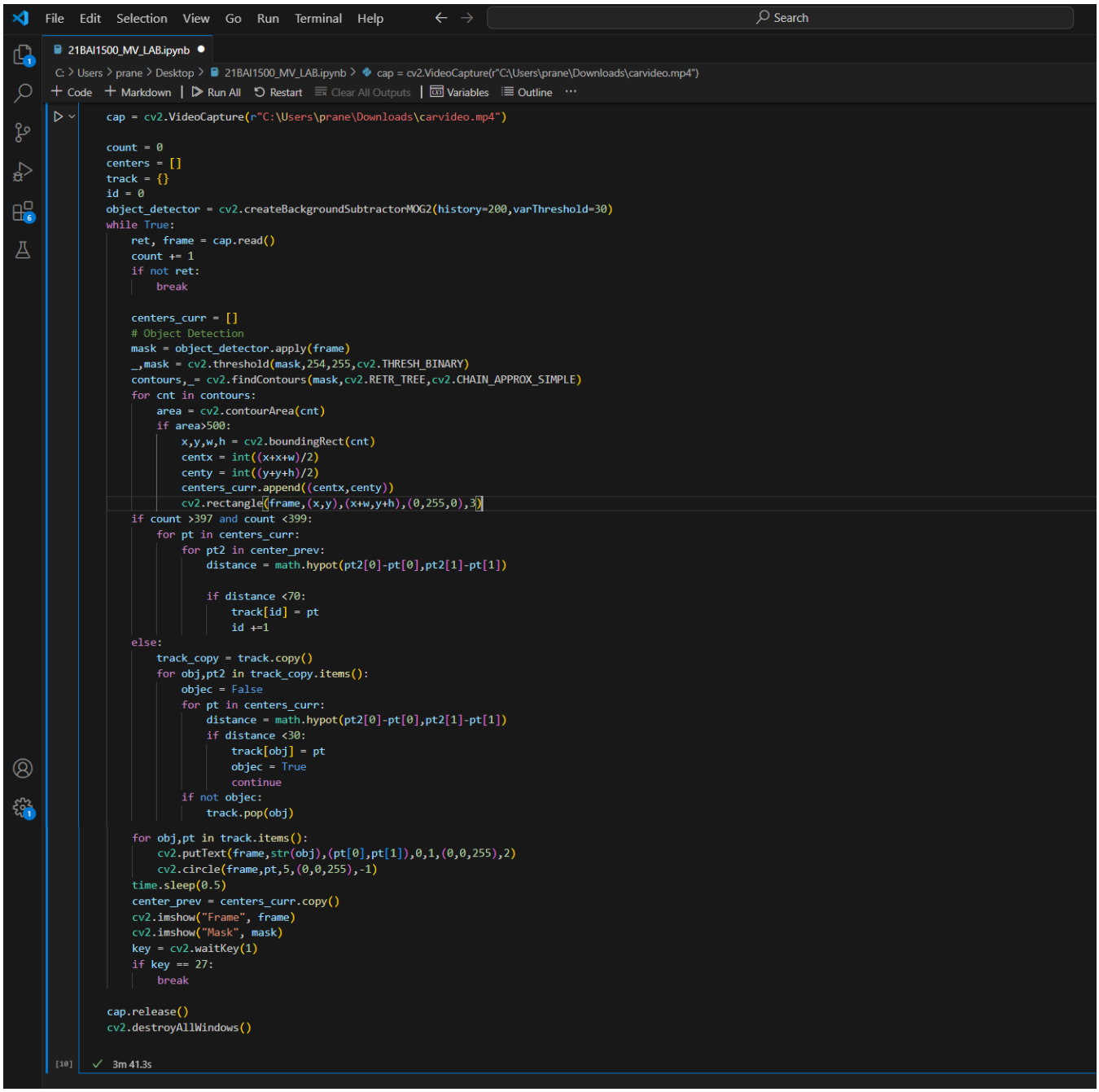
Task -3: Detect the objects and Find contours



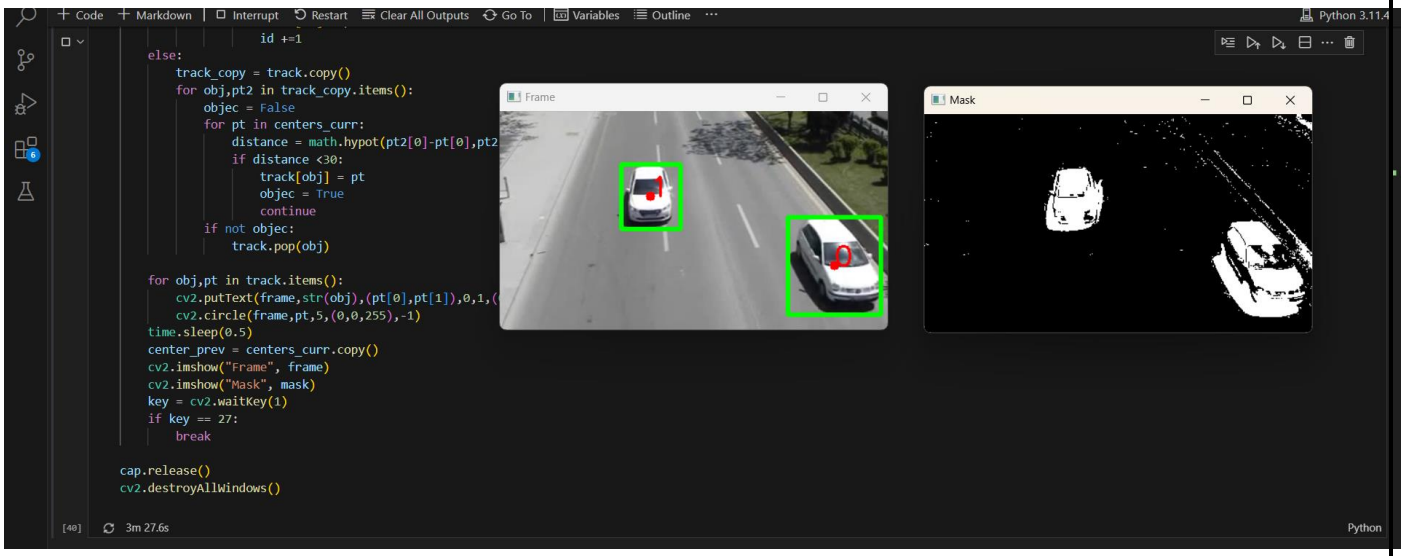
Task -4&5: Determine the centroid and assign unique object ids and finally track the objects.

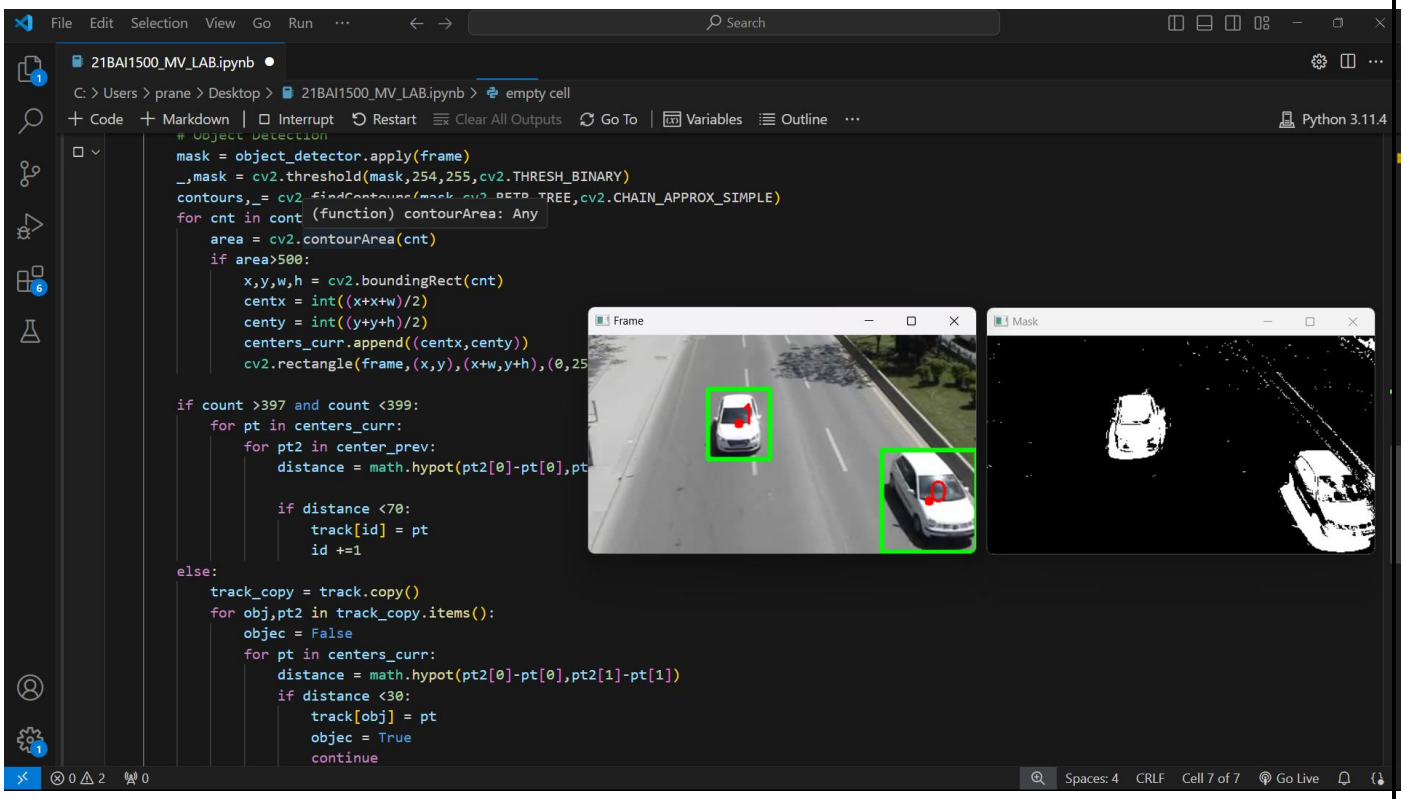
- Initially we detect the objects using contours, then make bounding boxes of them using boundingRect in cv2, then using the calculated box boundaries, calculate the centroid point x and y Update the centroids and store them in a list.
- During a single frame mark the found centroids and assign unique object ids to them. Following which from the next frame, calculate the distance between the marked centroid and the new centroid, if it is less than 30 then give the same id to it too.

- In this method we can track the object using centroid tracking.



```
cap = cv2.VideoCapture(r"C:\Users\prane\Downloads\carvideo.mp4")

count = 0
centers = []
track = {}
id = 0
object_detector = cv2.createBackgroundSubtractorMOG2(history=200,varThreshold=30)
while True:
    ret, frame = cap.read()
    count += 1
    if not ret:
        break

    centers_curr = []
    # Object Detection
    mask = object_detector.apply(frame)
    _,mask = cv2.threshold(mask,254,255,cv2.THRESH_BINARY)
    contours,_= cv2.findContours(mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
    for cnt in contours:
        area = cv2.contourArea(cnt)
        if area>500:
            x,y,w,h = cv2.boundingRect(cnt)
            centx = int((x+x+w)/2)
            centy = int((y+y+h)/2)
            centers_curr.append((centx,centy))
            cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),3)
    if count >397 and count <399:
        for pt in centers_curr:
            for pt2 in center_prev:
                distance = math.hypot(pt2[0]-pt[0],pt2[1]-pt[1])

                if distance <70:
                    track[id] = pt
                    id +=1
    else:
        track_copy = track.copy()
        for obj,pt2 in track_copy.items():
            objec = False
            for pt in centers_curr:
                distance = math.hypot(pt2[0]-pt[0],pt2[1]-pt[1])
                if distance <30:
                    track[obj] = pt
                    objec = True
                    continue
            if not objec:
                track.pop(obj)

    for obj,pt in track.items():
        cv2.putText(frame,str(obj),(pt[0],pt[1]),0,1,(0,0,255),2)
        cv2.circle(frame,pt,5,(0,0,255),-1)
    time.sleep(0.5)
    center_prev = centers_curr.copy()
    cv2.imshow("Frame", frame)
    cv2.imshow("Mask", mask)
    key = cv2.waitKey(1)
    if key == 27:
        break

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





Results: The given tasks have been done with the help of OpenCV and numpy libraries.

Conclusion: Python program have been created to detect the objects in a video using Contours and to track the detected objects with the help of Centroid Tracking method.