Exp 15: Count Vehicles in Image and Video

Aim : Automatic Counting of Vehicles in Images and Video Files Using OpenCV2

Theory : Video analysis separates moving objects (vehicles) from the background. It analyzes object shapes to focus on vehicles, then tracks them to count when they cross a virtual line. This technique helps automate traffic flow monitoring.

Algorithm :

1. **Read Video Frame:** Capture a frame from the video.
2. **Preprocess Image:** Convert the frame to grayscale , then Apply a Gaussian blur.
3. **Background Subtraction:** Use a background subtractor to separate moving objects.
4. **Morphological Operations:** Dilate the foreground objects and Perform morphological closing.
5. **Find Contours:** Identify the boundaries of foreground objects.
6. **Vehicle Filtering:** Loop through each contour , Calculate bounding box dimensions and Discard objects below minimum size criteria.
7. **Object Tracking:** For contours meeting size criteria , Calculate the center point of the bounding box and Append the center point to a tracking list.
8. **Counting Line:** Draw a virtual line on the frame.
9. **Vehicle Count:** Loop through tracked center points , If a center point crosses the line downwards, increment the counter. Remove the center point from the tracking list.
10. **Display Results:**Draw bounding boxes around detected vehicles , Display the current vehicle count and Show the processed image.

Program :

#install these Modules beforehand :

#pip install opencv-contrib-python

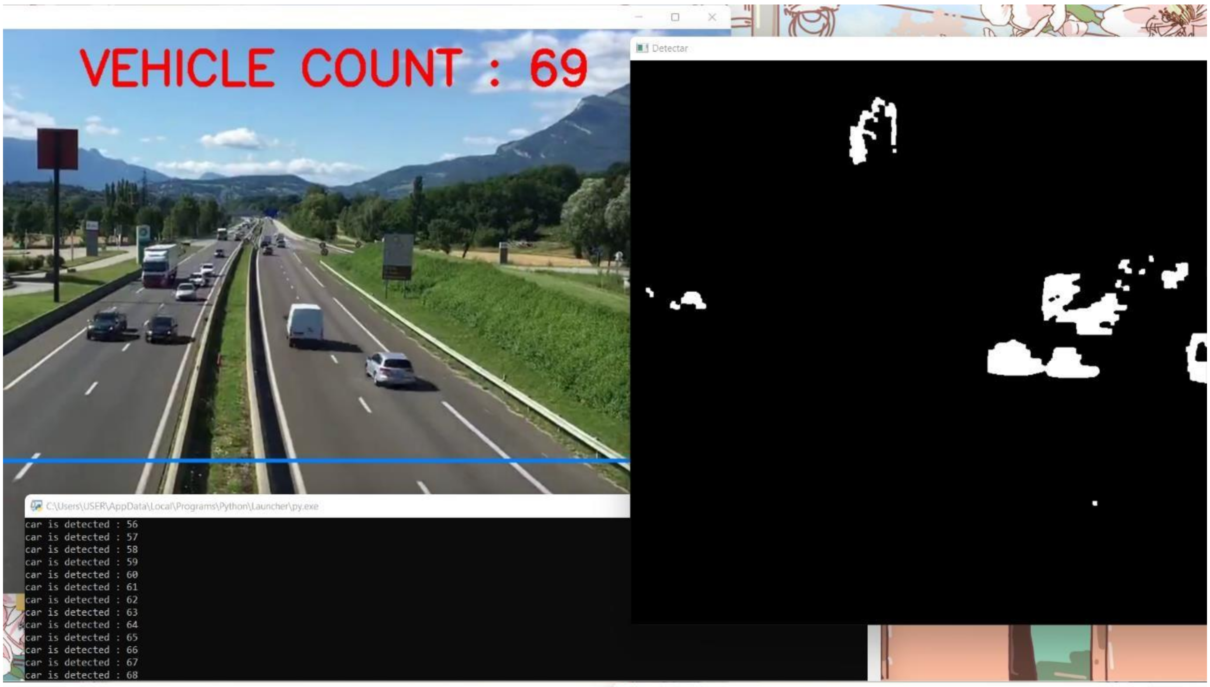
#pip install opencv-python

#pip install times

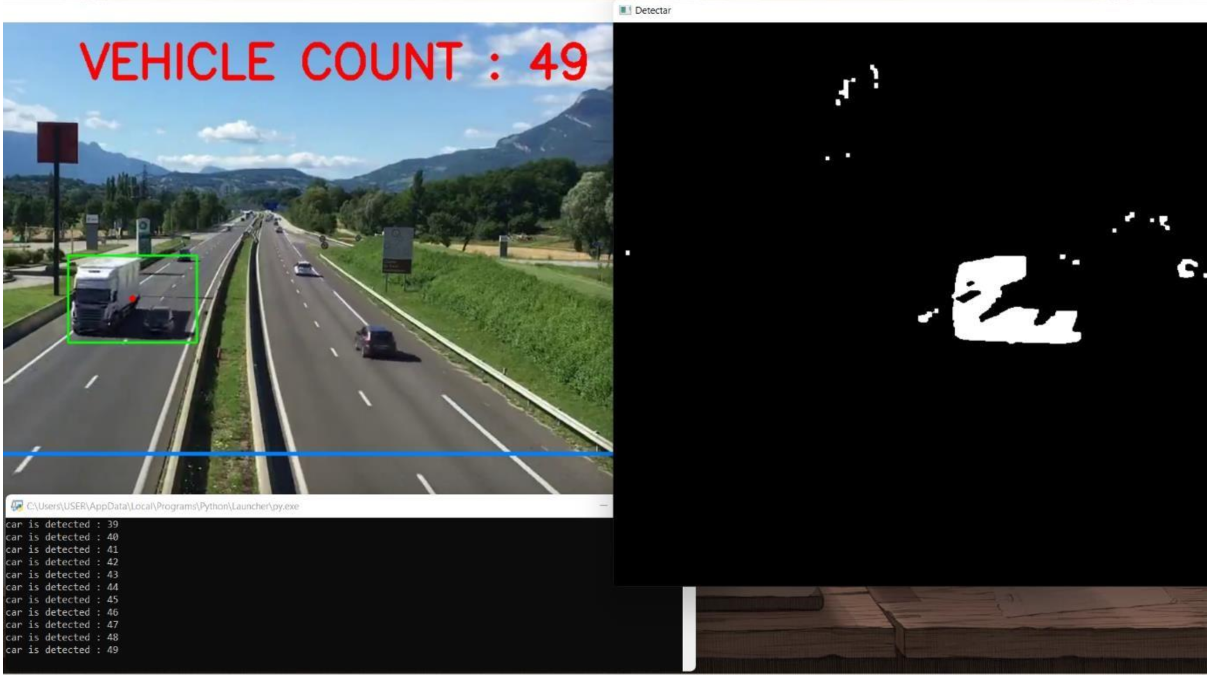
#pip install numpy

```
import    cv2    import
numpy as np from time
import sleep
 largura_min=80 #Largura minima do retangulo
altura_min=80 #Altura minima do retangulo

offset=6 #Erro permitido entre pixel

pos_linha=550 #Posição da linha de contagem

delay= 60 #FPS do vídeo
 detec =
[] carros=
0

def pega_centro(x, y, w, h):
x1 = int(w / 2)     y1 =
int(h / 2)     cx = x + x1
cy = y + y1     return cx,cy

cap = cv2.VideoCapture('video.mp4')
subtracao = cv2.bgsegm.createBackgroundSubtractorMOG()
 while
True:
    ret , frame1 = cap.read()
tempo = float(1/delay)
sleep(tempo)
    grey = cv2.cvtColor(frame1,cv2.COLOR_BGR2GRAY)
blur = cv2.GaussianBlur(grey,(3,3),5)     img_sub
= subtracao.apply(blur)     dilat =
cv2.dilate(img_sub,np.ones((5,5)))
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
dilatada = cv2.morphologyEx (dilat, cv2. MORPH_CLOSE , kernel)
dilatada = cv2.morphologyEx (dilatada, cv2. MORPH_CLOSE , kernel)

contorno,h=cv2.findContours(dilatada,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIM
PLE)

    cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (255,127,0),
3)      for(i,c) in
enumerate(contorno):          (x,y,w,h)
= cv2.boundingRect(c)
        validar_contorno = (w >= largura_min) and (h >= altura_min)
if not validar_contorno:
            continue
        cv2.rectangle(frame1,(x,y),(x+w,y+h),(0,255,0),2)
centro = pega_centro(x, y, w, h)         detec.append(centro)
        cv2.circle(frame1, centro, 4, (0, 0,255), -1)
         for (x,y) in detec:             if
y<(pos_linha+offset) and y>(pos_linha-offset):
                carros+=1
                cv2.line(frame1, (25, pos_linha), (1200, pos_linha),
(0,127,255), 3)
                detec.remove((x,y))
                print("car is detected : "+str(carros))

    cv2.putText(frame1, "VEHICLE COUNT : "+str(carros), (450, 70),
cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255),5)     cv2.imshow("Video
Original" , frame1)     cv2.imshow("Detectar",dilatada)
     if cv2.waitKey(1) ==
27:
        break
cv2.destroyAllWindows() cap.release()
```

Output:



```
car is detected : 56
car is detected : 57
car is detected : 58
car is detected : 59
car is detected : 60
car is detected : 61
car is detected : 62
car is detected : 63
car is detected : 64
car is detected : 65
car is detected : 66
car is detected : 67
car is detected : 68
```



```
car is detected : 39
car is detected : 40
car is detected : 41
car is detected : 42
car is detected : 43
car is detected : 44
car is detected : 45
car is detected : 46
car is detected : 47
car is detected : 48
car is detected : 49
```

Result : Hence Successfully Implemented the Program for Detection and Counting of Vehicles Using OpenCV2 .