

```

In [ ]: import numpy as np
import cv2
from time import sleep

min_width_rec = 80
min_height_rec = 80

offset = 6 # Allow error between pixel
delay = 60 # FPS to video
carros = 0 # counter for the number of detected vehicles

count_lines_pos = 550
detec = []
def central_handle(x,y,w,h): # this will calculate the center coordinate of a rectangle
    x1 = int(w/2)
    y1 = int(h/2)
    cx = x + x1
    cy = y + y1
    return cx, cy

# Enable web camera
cap = cv2.VideoCapture('video.mp4')
algo = cv2.bgsegm.createBackgroundSubtractorMOG()
#algo = cv2.createBackgroundSubtractorMOG2()
while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    blur = cv2.GaussianBlur(gray, (3,3),5)

    # Applying on each frame
    img_sub = algo.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)
    counterShape, h = cv2.findContours(dilatada, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

    cv2.line(frame, (25, count_lines_pos), (1200, count_lines_pos), (0,0,255),3)
    #Line(img, pt1, pt2, color,thickness)

    for (i, c) in enumerate(counterShape):
        (x,y,w,h) = cv2.boundingRect(c)
        validar_contorno = (w >=min_width_rec) and (h >=min_height_rec)

        if not validar_contorno:
            continue

        cv2.rectangle(frame, (x, y), (x+w, y+h),(0,255,0),2)
        cv2.putText(frame, "vehicle" + str(carros), (x, y-20), cv2.FONT_HERSHEY_TRIPLEX, 1, (255,244,0),2)
        # putText(img, text, org, fontFace, fontScale, color, thickness)

        center = central_handle(x,y,w,h)
        detec.append(center)
        cv2.circle(frame, center, 4, (255,100,100),-1)
        #circle(img, center, radius, color[, thickness)

    # Loop
    for (x,y) in detec:
        if y <(count_lines_pos + offset) and y >(count_lines_pos - offset):
            carros +=1
            cv2.line(frame, (25, count_lines_pos), (1200, count_lines_pos), (0,0,255),3)
            detec.remove((x,y))
            print("Vehicle is detected :" + str(carros))
    cv2.putText(frame, "Vehicle Count :" + str(carros), (450,70),cv2.FONT_HERSHEY_SIMPLEX, 2, (0,255,255), 5)
    # putText(img, text, org, fontFace, fontScale, color, thickness)

    # rectangle(img, pt1, pt2, color, thickness)

    cv2.imshow("Original Video with Vehicle Count", frame)

    if cv2.waitKey(1)==13:
        break
cap.release()
cv2.destroyAllWindows()

```

