```
import cv2
import numpy as np
from time import sleep

length_min = 80
height_min = 80

offset = 6  #
pos_linha = 550

delay = 60  # FPS do vídeo

detec = []
car_ros = 0
car_down=0


def pega_centro_1(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')##rtsp://<ip>:<port 554>/Streaming/Channels/<id>
<<2nd>--rtsp://username:password@/<ip>:<port 554>/Streaming/Channels/<id>
subtractor = 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)          ##gaussian method to blur the
background
    img_sub = subtractor.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_SIMPLE)

    cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (0, 127, 0), 3)
    for (i, c) in enumerate(contorno):
        (x, y, w, h) = cv2.boundingRect(c)
        validate_contorno = (w >= length_min) and (h >= height_min)
        if not validate_contorno:
            continue

        cv2.rectangle(frame1, (x, y), (x + w, y + h), (0, 255, 0), 2)
        centro = pega_centro_1(x, y, w, h)
        detec.append(centro)
        cv2.circle(frame1, centro, 4, (0, 0, 255), -1)
```

```
        print(centro)
        print(detec)
        for (x, y) in detec:
            if y < (pos_linha + offset) and y > (pos_linha - offset) and x<625: ##
centroid range for IN
                car_ros += 1
                cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (0, 127, 255), 3)
                detec.remove((x, y))
                print("car is detected : " + str(car_ros))
            elif y < (pos_linha + offset) and y > (pos_linha - offset) and x>625 and
x<=1200:  ## centroid range for OUT
                car_down += 1
                cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (0, 127, 255), 3)
                detec.remove((x, y))
                print("car goin out is detected : " + str(car_down))

    cv2.putText(frame1, "VEHICLE COUNT In: " + str(car_ros), (700, 80),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    cv2.putText(frame1, "VEHICLE COUNT out: " + str(car_down), (700, 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    cv2.putText(frame1, " Estimate:" + str(car_ros*50), (850, 150),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)
    cv2.imshow("Video Original", frame1)
    cv2.imshow("Detectar", dilatada)

    if cv2.waitKey(1) == 27:
        break

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