

Protokoll:

Inhaltsverzeichnis

-Raspi

-Server Backend

Raspi:

Beim Raspi wurde nicht Vieles gemacht, da das Programm von dem Autozähler nicht funktioniert hat. Sebastian Hager hat dann 2 Stunden mit dem Programm gekämpft, um den Fehler zu beheben.

Nun funktioniert es.

Server Backend:

Unser Plan war, die Date von dem Autozähler an dem Node-Server mit Hilfe des requests Modules (Programmsprache Python) zu schicken.

Damit haben Sebastian Lautischer und Abdulrahman Al Sabagh begonnen. Den Node-Server wurde in dem UE zum Laufen gebracht. Es fehlt nur das Posten der Daten. Da wir Probleme mit dem Raspi hatten, mussten wir das Python auf dem Laptop von Sebastian Lautischer installieren. Außerdem mussten wir dem PiP3 Kommando auf dem Laptop aktualisieren

Wir haben aber schon Python Code aber es funktioniert derzeit nur mit einem Video

```
main.py > ...
1  import cv2
2  import numpy as np
3  from time import sleep
4
5  largura_min=80 #Largura minima do retangulo
6  altura_min=80 #Altura minima do retangulo
7
8  offset=6 #Erro permitido entre pixel
9
10 pos_linha=580 #Posição da linha de contagem
11
12 delay= 60 #FPS do vídeo
13
14 detec = []
15 carros= 0
16
17
18 def pega_centro(x, y, w, h):
19     x1 = int(w / 2)
20     y1 = int(h / 2)
21     cx = x + x1
22     cy = y + y1
23     return cx,cy
24
25 cap = cv2.VideoCapture('video.mp4')
26 subtracao = cv2.bgsegm.createBackgroundSubtractorMOG()
27
28 while True:
29     ret , frame1 = cap.read()
30     tempo = float(1/delay)
31     sleep(tempo)
32     grey = cv2.cvtColor(frame1,cv2.COLOR_BGR2GRAY)
33     blur = cv2.GaussianBlur(grey,(3,3),5)
34     img_sub = subtracao.apply(blur)
35     dilat = cv2.dilate(img_sub,np.ones((5,5)))
36     kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
37     dilatada = cv2.morphologyEx (dilat, cv2. MORPH_CLOSE , kernel)
```

```
38 dilatada = cv2.morphologyEx (dilatada, cv2. MORPH_CLOSE , kernel)
39 contorno,h=cv2.findContours(dilatada,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
40
41 cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (255,127,0), 3)
42 for(i,c) in enumerate(contorno):
43     (x,y,w,h) = cv2.boundingRect(c)
44     validar_contorno = (w >= largura_min) and (h >= altura_min)
45     if not validar_contorno:
46         continue
47
48     cv2.rectangle(frame1,(x,y),(x+w,y+h),(0,255,0),2)
49     centro = pega_centro(x, y, w, h)
50     detec.append(centro)
51     cv2.circle(frame1, centro, 4, (0, 0,255), -1)
52
53     for (x,y) in detec:
54         if y<(pos_linha+offset) and y>(pos_linha-offset):
55             carros+=1
56             cv2.line(frame1, (25, pos_linha), (1200, pos_linha), (0,127,255), 3)
57             detec.remove((x,y))
58             print("car is detected : "+str(carros))
59
60 cv2.putText(frame1, "VEHICLE COUNT : "+str(carros), (450, 70), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255),5)
61 cv2.imshow("Video Original" , frame1)
62 cv2.imshow("Detector",dilatada)
63
64 if cv2.waitKey(1) == 27:
65     break
66
67 cv2.destroyAllWindows()
68 cap.release()
69
```