

0T.C.
BALIKESİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



ARAÇ YOĞUNLUK TAKİP SİSTEMİ

UMUT BUĞRA TER - 201913709019

BİTİRME PROJESİ RAPORU

Jüri Üyeleri : Dr. Öğr. Üyesi Gültekin KUVAT (Tez Danışmanı)

BALIKESİR, HAZİRAN- 2023

ETİK BEYAN

Balıkesir Üniversitesi Mühendislik Fakültesi Tez Yazım Kurallarına uygun olarak tarafımca hazırlanan “**Araç Yoğunluk Takip Sistemi**” başlıklı tezde;

- Tüm bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Kullanılan veriler ve sonuçlarda herhangi bir değişiklik yapmadığımı,
- Tüm bilgi ve sonuçları bilimsel araştırma ve etik ilkelere uygun şekilde sunduğumu,
- Yararlandığım eserlere atıfta bulunarak kaynak gösterdiğimi,

beyan eder, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ederim.

Umut Buğra TER

ÖZET

**ARAÇ YOĞUNLUK TAKİP SİSTEMİ
BİTİRME PROJESİ
UMUT BUĞRA TER
BALIKESİR ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ GÜLTEKİN KUVAT)
BALIKESİR, HAZİRAN - 2023**

Araç yoğunluk takip sistemi projesini yapmaktaki amacım, seçilen mobese kamerasındaki trafiğin hangi saatler arasında daha çok yoğun olduğunu anlayabilmektir.

İlk olarak araç sayma işlemini yapmak zorundaydım. Bunu şimdilik mobese kamerası üzerinden değil de normal dosyamızın içinde bulunan trafik kaydını gösteren bir video sayesinde yaptım. Daha sonra bu verileri kaydetmek için 'xlsxwriter' kütüphanesini kullandım ve grafikte göstermek için 'matplotlib.pyplot' kütüphanesini kullandım. Daha sonra canlı veri olan mobese kameralarından görüntü çekerek, yazdığım kodları mobese kamerasına uygun şekilde ayarlama yaptım. Araç sayma işlemleri bitince sıra araçların renklerini ayırt etmeyi yapmaya karar verdim. Maske oluşturarak bulmak istediğim renk aralığını vererek, o verdiğim renk aralığına sahip bölgeler belli olup, diğer bölgeler siyah renk ile gösteriliyor. Bu şekilde araç renk tespitini de yaptıktan sonra bunları tek bir projede birleştirerek, projemi tamamladım.

ANAHTAR KELİMELER:

Bilim Kod / Kodları :
Araç, Renk ayırımı, Mobese, Trafik Yoğunluğu

Sayfa Sayısı : 31

ABSTRACT

**VEHICLE DENSITY TRACKING SYSTEM
GRADUATION PROJECT
UMUT BUĞRA TER
BALIKESİR UNIVERSITY FACULTY OF ENGINEERING
COMPUTER ENGINEERING**

(SUPERVISOR: ASSIST. PROF. DR. GÜLTEKİN KUVAT)

BALIKESİR, JUNE - 2023

My purpose in making the vehicle density monitoring system project was to understand between which hours the traffic in the selected mobese camera was the most intense.

First I had to do vehicle counting. For now, I did this not through the security camera, but thanks to a video that shows the traffic protection in our regular file. It then covers the 'matplotlib.pyplot' library where it uses 'xlsx.writer' library for these data transfers and to show it in the graph. Then, I took images from live data security cameras and edited the codes I wrote according to security cameras. When the vehicle counting process was finished, I decided not to distinguish the next colors. Using the color space to place the mask, regions of that mesh with color space are highlighted, while other regions are shown in black. In this way, after seeing the vehicle color detection. I combined them into a single project and completed my project.

KEYWORDS:

Science Code / Codes :

Page Number : 31

Vehicle, Color Seperation, Security, Traffic Jam

İÇİNDEKİLER

Sayfa

| | |
|---|------------|
| ÖZET | i |
| ABSTRACT | ii |
| İÇİNDEKİLER..... | iii |
| ŞEKİL LİSTESİ | iv |
| 1. ARAÇ YOĞUNLUK TAKİP SİSTEMİ | 1 |
| 1.1 Araç Algılama ve Sayma İşlemi | 1 |
| 1.2 İnternet Üzerinden Canlı Veri Çekme | 3 |
| 1.3 Verileri Kaydetme ve Grafikselsel Olarak Gösterme | 6 |
| 1.4 Uzun Araçları Ayırt Etme..... | 9 |
| 1.5 Araç Renklerini Ayırt Etme..... | 12 |
| 2. PROJENİN GELİŞTİRME AŞAMASI..... | 16 |
| 3 SONUÇLAR..... | 20 |
| 4. KAYNAKLAR (IEEE) | 30 |
| ÖZGEÇMİŞ | 31 |

ŞEKİL LİSTESİ

| | Sayfa |
|--|--------------|
| Şekil 1.1: Araçları Algılama ve Sayma İşlemi Yapmamı Sağlayan Kod | 1 |
| Şekil 1.2: Araç Algılama ve Sayma İşlemi Yapan Kodun Çıktısı | 2 |
| Şekil 2.1: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 1..... | 3 |
| Şekil 2.2: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 2..... | 4 |
| Şekil 2.3: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 3..... | 4 |
| Şekil 2.4: Kopyalanan Bağlantının(URL) Koyulma Yeri..... | 5 |
| Şekil 2.5: Kopyalanan Bağlantı Sayesinde Canlı Çektiğimiz Mobese'nin Görüntüsü..... | 5 |
| Şekil 3.1: Gidiş ve Geliş (Çift Yönlü) İçin Sensör Oluşturma..... | 6 |
| Şekil 4.1: Verileri Kaydetmek İçin Kullandığım Kütüphane ve Oluşturduğum Sütun Adları | 7 |
| Şekil 4.2: Oluşturduğum Sütunlara Değer Atama İşlemi..... | 8 |
| Şekil 4.3: Grafiksel Olarak Göstermek İçin Yazdığım Kod | 8 |
| Şekil 4.4: Grafiksel Gösterimi..... | 9 |
| Şekil 5.1: Uzun Araçları Ayırt Etmek İçin Oluşturulan Değişken ve Diziler..... | 9 |
| Şekil 5.2: Uzun Araçların Sayısını Kaydetme | 10 |
| Şekil 5.3: Uzun ve Normal Araçları Ayırt Etme..... | 11 |
| Şekil 5.4: Uzun Araçların ve Normal Araçların Aynı Anda Geçtiği Fotoğraf | 11 |
| Şekil 6.1: İstenilen Rengin Renk Aralığını Girme | 12 |
| Şekil 6.2: Oluşturulan Maskeler Sayesinde İstenilen Renkteki Aracı Dikdörtgen İçine Alma | 13 |
| Şekil 6.3: Renk Kodlarının Uygulanmasıyla Oluşan Görüntü..... | 14 |
| Şekil 7.1: En Çok ve Az Sayıda Geçilen Saatleri Öğrenmek İçin Gerekli Değişkenler | 15 |
| Şekil 7.2: Dizi İçindeki En Büyük ve En Küçük Değişkenlerin İF Metoduyla Bulma | 15 |
| Şekil 8.1: Frameleri Renklerine Göre Ayır Ayır Oluşturma..... | 16 |
| Şekil 8.2: Oluşturulan Frame Sahip Olduğu Renk Koduna Göre Dikdörtgene Alma | 17 |
| Şekil 8.3: Frameleri Renklerine Göre Oranlama..... | 18 |
| Şekil 8.4: putText Fonksiyonun Kullanımı | 19 |
| Şekil 8.5: Kodun Çıktısı..... | 19 |

ARAÇ YOĞUNLUK TAKİP SİSTEMİ

1.1) Araç Algılama ve Sayma İşlemi

İlk olarak internetten araştırmalar sonucu bulduğum videolara ve internet sitelerine bakarak araçları algılama ve saydırma işlemi yaptım. Normal trafik olan bir videoyu indirerek işlemimizi başlattım. Daha sonra yapmak üzere mobese üzerinden canlı trafik kaydını çekme işlemi yapacağım.

```

yeniRenk.py  proje.py  mukemmel.py \  mukemmel1.py  sil.py  denemeRenk.py  mukemmel.py C:\yedek proje  deneme copy.py  deneme.py X
C:\Users\umutb\Desktop\yedek proje> deneme.py ...
1  import os
2  import cv2
3  import numpy as np
4  import time
5
6  Video_Okuyucu = cv2.VideoCapture("video1.mp4")
7  kernel = np.ones((5,5),np.uint8) #5x5 matris olusturup icine 1 degerleri attı
8
9
10 class Koordinat :
11     def __init__(self,x,y):
12         self.x = x
13         self.y = y
14
15 class Sensor :
16     def __init__(self,Koordinat1,Koordinat2,Kare_Genislilik,Kare_Yukseklilik) :
17         self.Koordinat1 = Koordinat1
18         self.Koordinat2 = Koordinat2
19         self.Kare_Genislilik = Kare_Genislilik
20         self.Kare_Yukseklilik = Kare_Yukseklilik
21         self.Maskenin_Alanı = abs(self.Koordinat2.x-self.Koordinat1.x)
22         self.Maske = np.zeros((Kare_Yukseklilik,Kare_Genislilik,1),np.uint8)*abs(self.Koordinat2.y-self.Koordinat1.y)
23         cv2.rectangle(self.Maske,(self.Koordinat1.x,self.Koordinat1.y),(self.Koordinat2.x,self.Koordinat2.y),(255),thickness=cv2.FILLED)
24         self.Durum = False
25         self.Algilanan_Arac_Sayisi = 0
26
27 ret, Kare = Video_Okuyucu.read()
28 Kesilmis_Kare = Kare[0:450, 0:450] #videonun boyutunu kesmeye yarıyor.
29 fgbg = cv2.createBackgroundSubtractorMOG2() #Arka planı silmeye yarıyor.
30 Sensor1 = Sensor(
31     Koordinat(10,415),
32     Koordinat(360,420),
33     Kesilmis_Kare.shape[0],
34     Kesilmis_Kare.shape[1])
35
36 font=cv2.FONT_HERSHEY_TRIPLEX
37
38
39 while 1:
40     ret, Kare = Video_Okuyucu.read()
41
42     Kesilmis_Kare = Kare[0:450, 0:450]
43     Arka_Planı_Silinmis_Kare = fgbg.apply(Kesilmis_Kare)
44
45     #morphological transform sitedesinden aldık. videoyu iyileştirmek için kullandık. Orada birşürü fonksiyon var
46     Arka_Planı_Silinmis_Kare = cv2.morphologyEx(Arka_Planı_Silinmis_Kare, cv2.MORPH_OPEN, kernel)
47     ret,Arka_Planı_Silinmis_Kare=cv2.threshold(Arka_Planı_Silinmis_Kare,127,255,cv2.THRESH_BINARY) #patlamaları yok etti
48
49
50     #contours lar sayesinde objeleri ayırma, kare içine alma
51     cnts, hierarchy = cv2.findContours(Arka_Planı_Silinmis_Kare,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

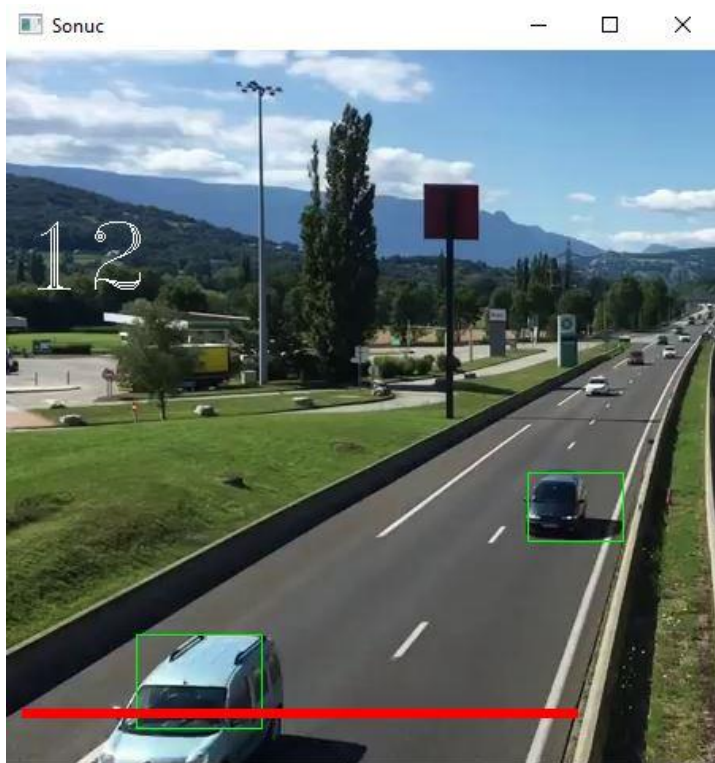
Şekil Error! No text of specified style in document..1: Araçları Algılama ve Sayma İşlemi Yapmamı Sağlayan Kod

```

46 Arka_Plani_Silinmis_Kare = cv2.morphologyEx(Arka_Plani_Silinmis_Kare, cv2.MORPH_OPEN, kernel)
47 ret,Arka_Plani_Silinmis_Kare=cv2.threshold(Arka_Plani_Silinmis_Kare,127,255,cv2.THRESH_BINARY) #patlamaları yok etti
48
49
50 #contours lar sayesinde objeleri ayırma, kare içine alma
51 cnts, hierarchy = cv2.findContours(Arka_Plani_Silinmis_Kare,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
52
53 Sonuc_Kesilmis_Kare.copy()
54 Doldurulmus_Resim = np.zeros((Kesilmis_Kare.shape[0],Kesilmis_Kare.shape[1],1),np.uint8)
55 for cnt in cnts:
56     x, y, w, h = cv2.boundingRect(cnt)
57     if (w>30 and h>30): #küçük küçük noktalarıda kare aldığı için boyut sınırı koyduk
58         # (Video, sol üst köşe,sag alt köşe,renk,çizgi kalınlığı) Aracı kare içine almaya yarıyor.
59         cv2.rectangle(Sonuc,(x,y),(x+w,y+h),(0,255,0),thickness=1)
60         cv2.rectangle(Doldurulmus_Resim,(x,y),(x+w,y+h),255,thickness=cv2.FILLED) #karenin içini dolduruyor
61
62
63 Sensor1_Maske_Sonuc = cv2.bitwise_and(Doldurulmus_Resim,Doldurulmus_Resim, mask=Sensor1_Maske)
64 Sensor1_Beyaz_Piksel_Sayisi = np.sum(Sensor1_Maske_Sonuc==255)
65 Sensor1_Oran = Sensor1_Beyaz_Piksel_Sayisi / Sensor1.Maskenin_Alani
66
67 # if(Sensor1_Oran>0):
68 #     print(Sensor1_Oran)
69
70 # Araba kırmızı çizgi sensörünün altından geçiyorsa, kırmızı çizgi sensörü yeşil renge geçecektir.
71 if (Sensor1_Oran>=0.9 and Sensor1_Oran<3.1 and Sensor1.Durum==False):
72     cv2.rectangle(Sonuc,(Sensor1.Koordinat1.x,Sensor1.Koordinat1.y),(Sensor1.Koordinat2.x,Sensor1.Koordinat2.y),(0,255,0),cv2.FILLED)
73     Sensor1.Durum = True;
74     Sensor1.Algılanan_Arac_Sayisi += 1
75
76 elif (Sensor1_Oran<0.9 and Sensor1.Durum==True) :
77     cv2.rectangle(Sonuc,(Sensor1.Koordinat1.x,Sensor1.Koordinat1.y),(Sensor1.Koordinat2.x,Sensor1.Koordinat2.y),(0,0,255),cv2.FILLED)
78     Sensor1.Durum = False;
79 else:
80     cv2.rectangle(Sonuc,(Sensor1.Koordinat1.x,Sensor1.Koordinat1.y),(Sensor1.Koordinat2.x,Sensor1.Koordinat2.y),(0,0,255),cv2.FILLED)
81
82 cv2.putText(Sonuc,str(Sensor1.Algılanan_Arac_Sayisi),(Sensor1.Koordinat1.x,150),font,2,(255,255,255))
83
84 #cv2.imshow("doldurulmus",Doldurulmus_Resim)
85 #cv2.imshow("maske",Sensor1_Maske)
86 #cv2.imshow("sensörlü maske sonuc",Sensor1_Maske_Sonuc)
87 cv2.imshow("Sonuc",Sonuc)
88
89
90
91 k=cv2.waitKey(30) & 0xFF #esc ye bastığında program kapanır
92 if k == 27:
93     break
94
95 Video_Okuyucu.release() #dosyayı serbest bırakma, sürekli tutmana kısaca kapatmak gibi
96 cv2.destroyAllWindows() #opencv pencerelerini yok ediyoruz

```

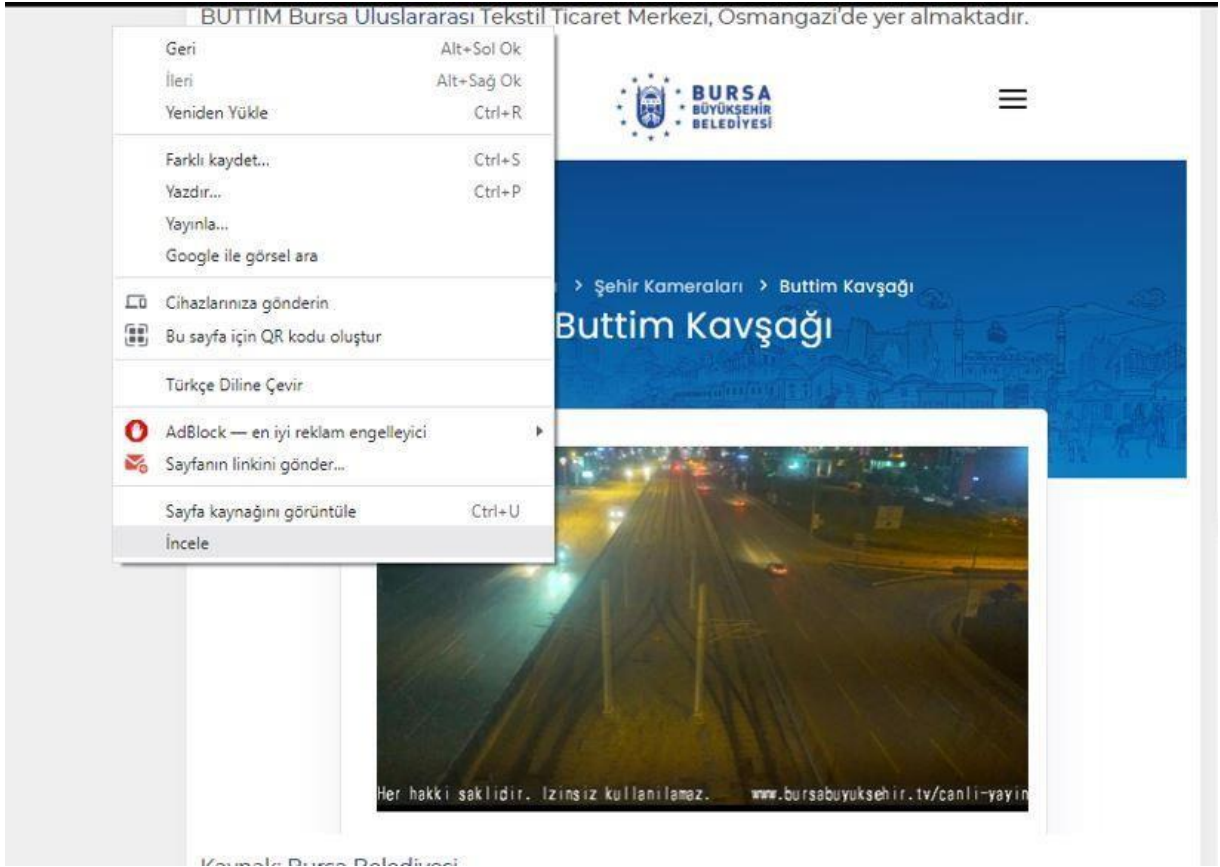
Şekil 1.1: (devam)



Şekil 1.2: Araç Algılama ve Sayma İşlemi Yapan Kodun Çıktısı

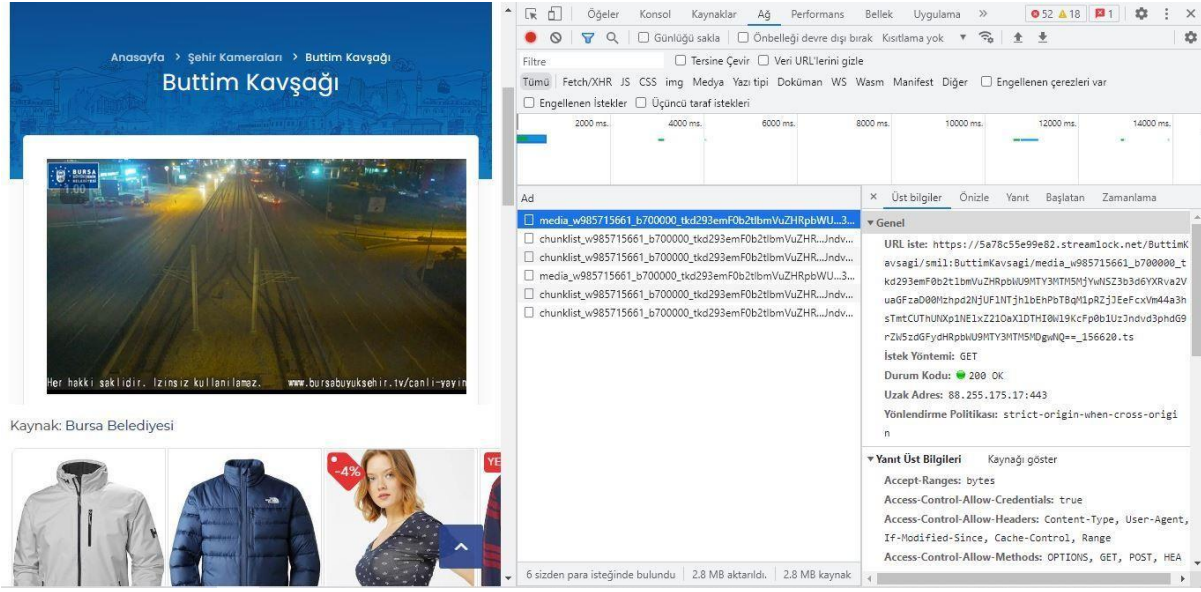
1.2) İnternet Üzerinden Canlı Veri Çekme

Bunu tamamladıktan sonra sıra Mobese kamerası üzerinden canlı trafik akışını çekmede. Canlı Mobese yayını çekebilmek için ilk olarak o siteye giriş yapıyoruz. Ben Bursa Buttım Kavşağını çekmeye karar verdim. Çünkü hem gelen ve giden araçları görebiliyoruz ve görüntü kalitesi diğer kameralara göre daha net olduğu için. Uzun araştırmalarım sonucu bir URL bağlantısı bulmamız gerektiğini öğrendim. Bu bağlantıyı bulabilmek için ilk önce o web sitesi ekranına sağ tıkladıktan sonra incele yerine basıyoruz ve karşımıza ekran çıkıyor.

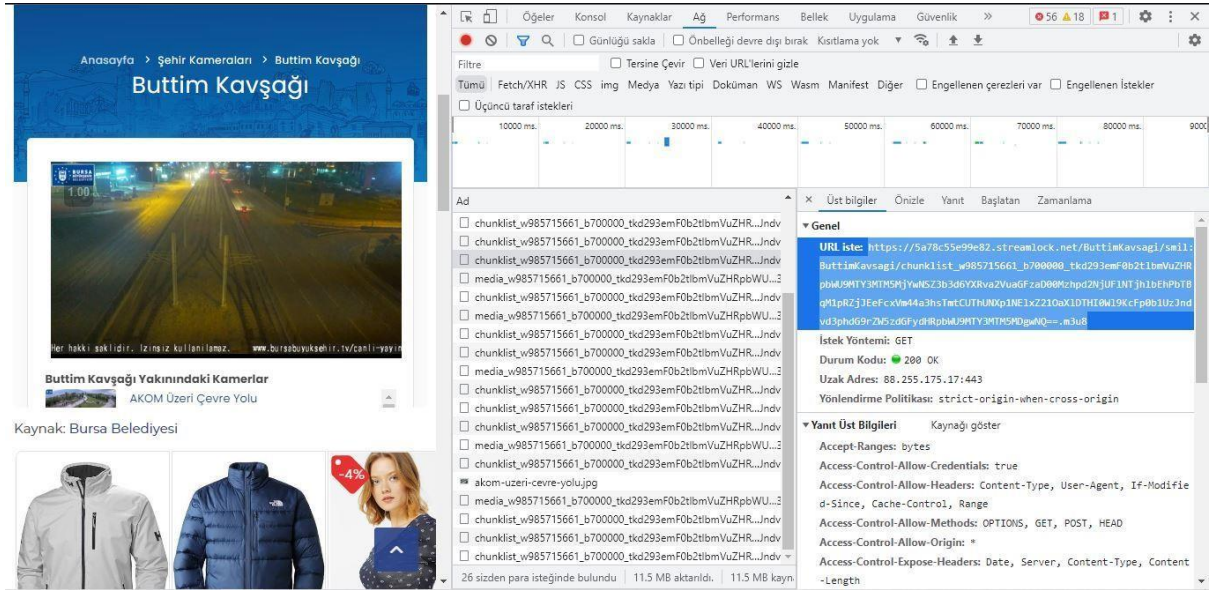


Şekil 2.1: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 1

Bu ekranda Ağ kısmına geliyoruz. Ağ'a tıkladıktan sonra Ad isminde ve o Ad ismine ait özellikleri içeren bir ekran karşımıza çıkıyor. Burada Ad isminde bulunan yerde bağlantılar var ve bu bağlantılara tıkladığımızda bu bağlantıya ait özellikler çıkıyor. Burada URL uzantısı (.m3m8) veya (.ts) olan bağlantıları almak zorunda olduğumuzu öğrendim. Ve uzantısı .m3m8 olan .ts olan bağlantıları kopyaladım.



Şekil 2.2: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 2

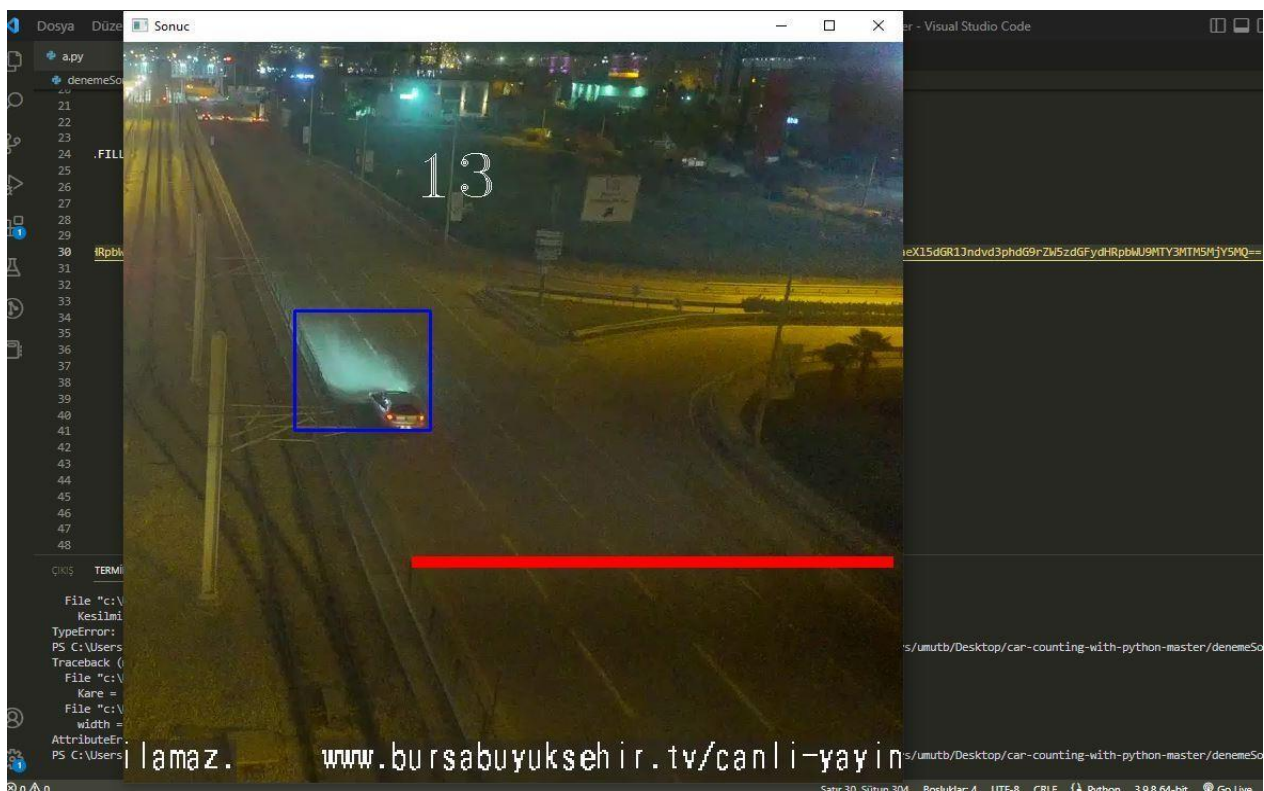


Şekil 2.3: Mobese Kamera Kayıtlarının Olduğu Siteden Veri Çekme Adımları 3

Kopyaladığım bağlantıyla ilgili ne yapmam gerektiğine dair pek bir fikrim yoktu fakat onu da uzun araştırmalarım sonucu, normal dosyamızda bulunan video değişkenine atamamızın yeteceğini buldum.

[illegible]

Şekil 2.4: Kopyalanan Bağlantının(URL) Koyulma Yeri



Şekil 2.5: Kopyalanan Bağlantı Sayesinde Canlı Çektiğimiz Mobese'nin Görüntüsü

1.3) Verileri Kaydetme ve Grafiksel Olarak Gösterme

Canlı olarak verimizi de çekebildiğimize göre sıra bunu görsele yani grafik üzerine dökmekte ve bu verileri excel dosyasına kaydetmekte. Bursa Buttim Kavşağında sadece tek yönden araçlar gelen olmadığı için, yani hem giden araç ve gelen araç olduğu için yeni bir sensör oluşturup diğer taraf içinde araç sayma işlemini yapmalıyız. İlk olarak yeni bir sensör üretilip onları ayarlayacağım.

```
72 Sensor1 = Sensor(  
73     Koordinat(800, Kesilmis_Kare.shape[1] - 830),  
74     Koordinat(1200, Kesilmis_Kare.shape[1] - 820),  
75     Kesilmis_Kare.shape[0],  
76     Kesilmis_Kare.shape[1])  
77  
78 Sensor2 = Sensor(  
79     Koordinat(0, Kesilmis_Kare.shape[1] - 840),  
80     Koordinat(275, Kesilmis_Kare.shape[1] - 830),  
81     Kesilmis_Kare.shape[0],  
82     Kesilmis_Kare.shape[1])  
83  
84 kernel=np.ones((5,5),np.uint8)  
85 font=cv2.FONT_HERSHEY_TRIPLEX  
86  
87  
88  
89 while (1):  
90     ret, Kare = Video_Okuyucu.read()  
91     Kare = rescale(Kare)  
92     # resize frame  
93     Kesilmis_Kare= Kare  
94     # make morphology for frame  
95     Arka_Plani_Silinmis_Kare=fgbg.apply(Kesilmis_Kare)  
96     Arka_Plani_Silinmis_Kare=cv2.morphologyEx(Arka_Plani_Silinmis_Kare,cv2.MORPH_OPEN,kernel)  
97     ret,Arka_Plani_Silinmis_Kare=cv2.threshold(Arka_Plani_Silinmis_Kare,127,255,cv2.THRESH_BINARY)  
98  
99     # detect moving anything  
100     cnts,_=cv2.findContours(Arka_Plani_Silinmis_Kare,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)  
101     Sonuc=Kesilmis_Kare.copy()  
102  
103     Doldurulmus_Resim=np.zeros((Kesilmis_Kare.shape[0],Kesilmis_Kare.shape[1], 1), np.uint8)  
104  
105     # detect moving anything with loop  
106     for cnt in cnts:  
107         x,y,w,h=cv2.boundingRect(cnt)  
108         if(w>30 and h>30):  
109             cv2.rectangle(Sonuc,(x,y),(x+w,y+h),(255,0,0),thickness=2)  
110             cv2.rectangle(Doldurulmus_Resim,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)  
111  
112     # detect whether there is car via bitwise_and  
113     mask1=np.zeros((Doldurulmus_Resim.shape[0],Doldurulmus_Resim.shape[1],1),np.uint8)  
114     Sensor1_Maske_Sonuc=cv2.bitwise_or(Doldurulmus_Resim,Doldurulmus_Resim,mask=Sensor1.Maske)  
115     Sensor1_Beyaz_Piksel_Sayisi=np.sum(Sensor1_Maske_Sonuc==255)  
116     Sensor2_Maske_Sonuc = cv2.bitwise_or(Doldurulmus_Resim,Doldurulmus_Resim,mask=Sensor2.Maske)  
117     Sensor2_Beyaz_Piksel_Sayisi=np.sum(Sensor2_Maske_Sonuc==255)  
118
```

Şekil 3.1: Gidiş ve Geliş (Çift Yönlü) İçin Sensör Oluşturma

```

124 #print(Sensor2_Oran)
125
126 if (Sensor1_Oran>=0.5 and Sensor1.Durum==False):
127     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
128                   (0,255, 0), thickness=cv2.FILLED)
129     Sensor1.Durum = True
130 elif (Sensor1_Oran>1 and Sensor1_Oran<=4 and Sensor1.Durum==True) :
131     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
132                   (0, 0,255), thickness=cv2.FILLED)
133     Sensor1.Durum = False
134     Sensor1.Algılanan_Arac_Sayisi+=1
135 elif (Sensor1_Oran>5.5 and Sensor1_Oran<=7.5 and Sensor1.Durum==True) :
136     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
137                   (0, 255,255), thickness=cv2.FILLED)
138     Sensor1.Durum = False
139 else :
140     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
141                   (0, 0, 255), thickness=cv2.FILLED)
142
143
144 if (Sensor2_Oran>=0.5 and Sensor2.Durum==False):
145     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
146                   (0,255, 0), thickness=cv2.FILLED)
147     Sensor2.Durum = True
148 elif (Sensor2_Oran>1 and Sensor2_Oran<=4 and Sensor2.Durum==True) :
149     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
150                   (0, 0,255), thickness=cv2.FILLED)
151     Sensor2.Durum = False
152     Sensor2.Algılanan_Arac_Sayisi+=1
153 elif (Sensor2_Oran>5.45 and Sensor2_Oran<=7.2 and Sensor2.Durum==True) :
154     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
155                   (0, 255,255), thickness=cv2.FILLED)
156     Sensor2.Durum = False
157 else :
158     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
159                   (0, 0, 255), thickness=cv2.FILLED)
160
161
162 cv2.putText(Sonuc, str(Sensor1.Algılanan_Arac_Sayisi), (Sensor1.Koordinat1.x,150), font, 2, (255,255,255))
163 cv2.putText(Sonuc, str(Sensor2.Algılanan_Arac_Sayisi), (Sensor2.Koordinat1.x,50), font, 2, (255,255,255))
164
165

```

Şekil 3.1: (devamı)

Diğer taraftan gelen araçlar için sensör oluşumu tamam. Sıra excele kaydetmekte ve grafik üzerinden göstermekte. Excel'e kaydetmek için ilk olarak 'xlsxwriter' kütüphanesini eklememiz gerekiyor. Grafiksel olarak görmek için de 'matplotlib.pyplot' kütüphanesini ekledik. Kaydetmek istediğimiz verileri aşağıdaki fotoğrafta gibi yapıyoruz. Böylelikle verilerimizi excel dosyasına kaydetmiş olduk.

```

4 import matplotlib.pyplot as plt
5 import xlsxwriter
6
7 planWorkBook = xlsxwriter.Workbook('veri.xlsx') #kaydedilecek dosyanın adı
8 planSheet = planWorkBook.add_worksheet("sayfaAdi")
9 planSheet.write("A1", "GECEN ZAMAN") # 1. sütunun adı
10 planSheet.write("B1", "GELEN ARAC SAYISI") # 2. sütunun adı
11 planSheet.write("C1", "GİDEN ARAC SAYISI") # 3. sütunun adı
12

```

Şekil 4.1: Verileri Kaydetmek İçin Kullandığım Kütüphane ve Oluşturduğum Sütun Adları

```
# k=cv2.waitKey(1000) & 0xff #esc ye bastığında program kapanır
if cv2.waitKey(60) == ord("q"):
    print(gecen_zaman)
    print(gelen_arac_sayisi)
    print(giden_arac_sayisi)
    print(gelen_uzun_arac_sayisi)
    print(giden_uzun_arac_sayisi)
    for i in range(0,len(gecen_zaman)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,0,gecen_zaman[i]) # (satır sırası, sütun sırası, değer) zaman sütunu 0. sütunda kaldığı için sadece satır.

    for i in range(0,len(gelen_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,1,gelen_arac_sayisi[i]) # Gelen araç sütunu 1. sütunda kaldığı için sadece satırları arttırdım

    for i in range(0,len(giden_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,2,giden_arac_sayisi[i]) # Giden araç sütunu 2. sütunda kaldığı için sadece satırları arttırdım
```

Şekil 4.2: Oluşturduğum Sütunlara Değer Atama İşlemi

Grafik olarak belirtmek için ilk olarak x-eksenine zamanı vermemiz gerekiyor ki y-eksenine ise geçen araçların sayısını girmemiz lazım ki hangi aralıklarda kaç tane aracın geçtiğini anlayalım. Hem gelen ve giden araçlar olduğu için 2 tane grafik oluşturacağım. Her ikisi içinde grafiği görelim.

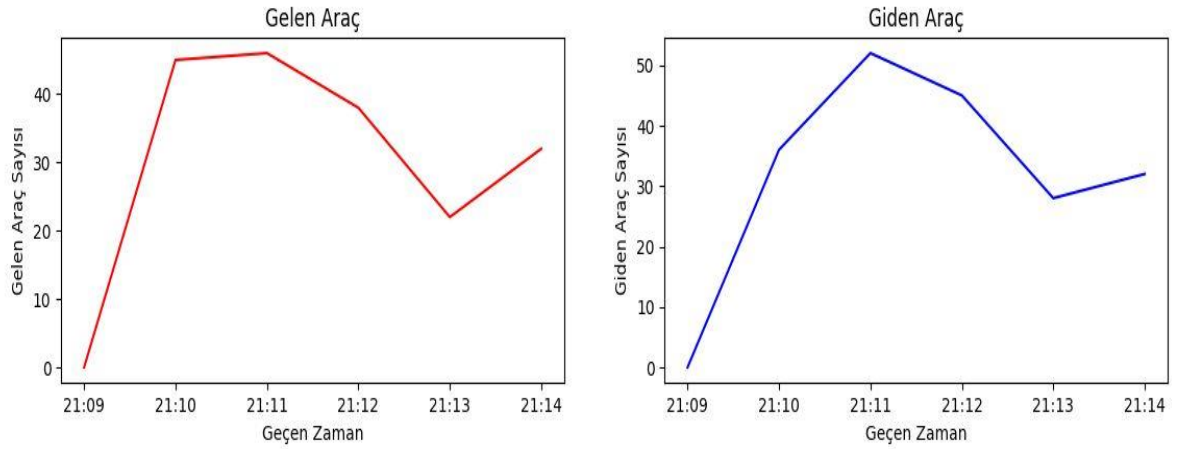
```
planWorkBook.close()
plt.figure(figsize=(20,10))

plt.subplot(2,2,1)
plt.plot(gecen_zaman,gelen_arac_sayisi,color="r")
plt.xlabel("Geçen Zaman")
plt.ylabel("Gelen Araç Sayısı")
plt.title("Gelen Araç")

plt.subplot(2,2,2)
plt.plot(gecen_zaman,giden_arac_sayisi,color="blue")
plt.xlabel("Geçen Zaman")
plt.ylabel("Giden Araç Sayısı")
plt.title("Giden Araç")
plt.show()
break
```

Şekil 4.3: Grafiksel Olarak Göstermek İçin Yazdığım Kod

Figure 1



Şekil 4.4: Grafiksel Gösterimi

1.4) Uzun Araçları Ayırt Etme

Şimdiye kadar tüm araçları aynı sınıfta saymıştık fakat artık şehirlerarası otobüsleri, kamyonları vs. ayrı bir kategori(gelen_uzun_arac_sayisi/giden_uzun_arac_sayisi), normal araçları zaten tanımlı olan (gelen_arac_sayisi/giden_arac_sayisi) bir kategoride tanıyacağız. Küçük motosikletleri, bisikletlerin çoğunu bilgisayar algılayamadığı için onları yok saydık. Onlar için ayrı bir kategori oluşturmam.

```
class Sensor:
    def __init__(self,Koordinat1,Koordinat2,Kare_Genislik,Kare_Yukseklilik):
        self.Koordinat1 = Koordinat1
        self.Koordinat2 = Koordinat2
        self.Kare_Genislik = Kare_Genislik
        self.Kare_Yukseklilik = Kare_Yukseklilik
        self.Maskenin_Alani = abs(self.Koordinat2.x-self.Koordinat1.x)
        self.Maske=np.zeros((Kare_Genislik,Kare_Yukseklilik,1),np.uint8)*abs(self.Koordinat2.y-self.Koordinat1.y)
        cv2.rectangle(self.Maske,(self.Koordinat1.x,self.Koordinat1.y),(self.Koordinat2.x,self.Koordinat2.y),(255),thickness=cv2.FILLED)
        self.Durum=False
        self.Algilanan_Arac_Sayisi=0
        self.Algilanan_Uzun_Arac_Sayisi=0

gelen_arac_sayisi = []
giden_arac_sayisi = []
gelen_uzun_arac_sayisi = []
giden_uzun_arac_sayisi = []
gecen_zaman = []
gelen_arac_sayisi.append(0)
giden_arac_sayisi.append(0)
gelen_uzun_arac_sayisi.append(0)
giden_uzun_arac_sayisi.append(0)
```

Şekil 5.1: Uzun Araçları Ayırt Etmek İçin Oluşturulan Değişken ve Diziler

Yeni bir kategori eklediğimiz için onları da veri tabanına kaydetmek lazım. Tekrar gelen/giden uzun araçların sayısını tutmak için yeni sütun oluşturdum. Bu sütunların değerini atmak için for döngüsü kullandım

```
if now < video_bölme_bitis_zamanı : # video_bölme_bitis_zamanı 'na aralığımızı ekledik ve zaman ona yaklaştıkça sürekli arttırıyoruz.
    cv2.imshow("Sonuc",Sonuc)
    now = datetime.now()
elif video_bölme_bitis_zamanı <= now: #%%H:%%M:%%S
    zaman = video_bölme_bitis_zamanı.strftime("%H:%M") #datetime string türüne dönüştürüldü
    geçen_zaman.append(zaman) #diziye aktarıldı
    video_bölme_bitis_zamanı += timedelta(seconds=aralıkT0saniye)
    gelen_arac_sayisi.append(Sensor2.Algılanan_Arac_Sayisi)
    giden_arac_sayisi.append(Sensor1.Algılanan_Arac_Sayisi)
    gelen_uzun_arac_sayisi.append(Sensor2.Algılanan_Uzun_Arac_Sayisi)
    giden_uzun_arac_sayisi.append(Sensor1.Algılanan_Uzun_Arac_Sayisi)
    Sensor1.Algılanan_Arac_Sayisi = 0
    Sensor2.Algılanan_Arac_Sayisi = 0
    Sensor1.Algılanan_Uzun_Arac_Sayisi = 0
    Sensor2.Algılanan_Uzun_Arac_Sayisi = 0

k=cv2.waitKey(60) & 0xff #esc ye bastığında program kapanır
if k == 27 :
    print(geçen_zaman)
    print(gelen_arac_sayisi)
    print(giden_arac_sayisi)
    print(gelen_uzun_arac_sayisi)
    print(giden_uzun_arac_sayisi)
    for i in range(0,len(geçen_zaman)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,0,geçen_zaman[i]) # (sıra sırası, sütun sırası, değer) zaman sütunu 0. sütunda kaldığı için sadece satırları arttırdım

    for i in range(0,len(gelen_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,1,gelen_arac_sayisi[i]) # Gelen araç sütunu 1. sütunda kaldığı için sadece satırları arttırdım

    for i in range(0,len(gelen_uzun_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,2,gelen_uzun_arac_sayisi[i]) # Gelen uzun araç sütunu 2. sütunda kaldığı için sadece satırları arttırdım

    for i in range(0,len(giden_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,3,giden_arac_sayisi[i]) # Giden araç sütunu 3. sütunda kaldığı için sadece satırları arttırdım

    for i in range(0,len(giden_uzun_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
        planSheet.write(i+1,4,giden_uzun_arac_sayisi[i]) # Giden uzun araç sütunu 4. sütunda kaldığı için sadece satırları arttırdım
```

Şekil 5.2: Uzun Araçların Sayısını Kaydetme

Sensörden araçlar geçerken, orada algıladığım orana göre onları ayrı kategorilere attım ve uzun araçların geçtiğini anlamamız için, normal araçlar geçerken yeşil olan sensörü sarı renge dönüştürdüm.


```

#print(Sensor2_Oran)

if (Sensor1_Oran>=0.5 and Sensor1.Durum==False):
    cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
        (0,255, 0,)), thickness=cv2.FILLED)
    Sensor1.Durum = True
elif (Sensor1_Oran>1 and Sensor1_Oran<=4 and Sensor1.Durum==True) :
    cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
        (0, 0,255), thickness=cv2.FILLED)
    Sensor1.Durum = False
    Sensor1.Algılanan_Arac_Sayisi+=1
elif (Sensor1_Oran>5.5 and Sensor1_Oran<=7.5 and Sensor1.Durum==True) :
    cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
        (0, 255,255), thickness=cv2.FILLED)
    Sensor1.Durum = False
    Sensor1.Algılanan_Uzun_Arac_Sayisi += 1
else :
    cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
        (0, 0, 255), thickness=cv2.FILLED)

if (Sensor2_Oran>=0.5 and Sensor2.Durum==False):
    cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
        (0,255, 0,)), thickness=cv2.FILLED)
    Sensor2.Durum = True
elif (Sensor2_Oran>1 and Sensor2_Oran<=4 and Sensor2.Durum==True) :
    cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
        (0, 0,255), thickness=cv2.FILLED)
    Sensor2.Durum = False
    Sensor2.Algılanan_Arac_Sayisi+=1
elif (Sensor2_Oran>5.45 and Sensor2_Oran<=7.2 and Sensor2.Durum==True) :
    cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
        (0, 255,255), thickness=cv2.FILLED)
    Sensor2.Durum = False
    Sensor2.Algılanan_Uzun_Arac_Sayisi += 1
else :
    cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
        (0, 0, 255), thickness=cv2.FILLED)

cv2.putText(Sonuc,str(Sensor1.Algılanan_Arac_Sayisi),(Sensor1.Koordinat1.x,150),font,2,(255,255,255))
cv2.putText(Sonuc,str(Sensor1.Algılanan_Uzun_Arac_Sayisi),(Sensor1.Koordinat1.x,250),font,2,(255,255,255))
cv2.putText(Sonuc,str(Sensor2.Algılanan_Arac_Sayisi),(Sensor2.Koordinat1.x,150),font,2,(255,255,255))
cv2.putText(Sonuc,str(Sensor2.Algılanan_Uzun_Arac_Sayisi),(Sensor2.Koordinat1.x,250),font,2,(255,255,255))

```

Şekil 5.3: Uzun ve Normal Araçları Ayırt Etme



Şekil 5.4: Uzun Araçların ve Normal Araçların Aynı Anda Geçtiği Fotoğraf

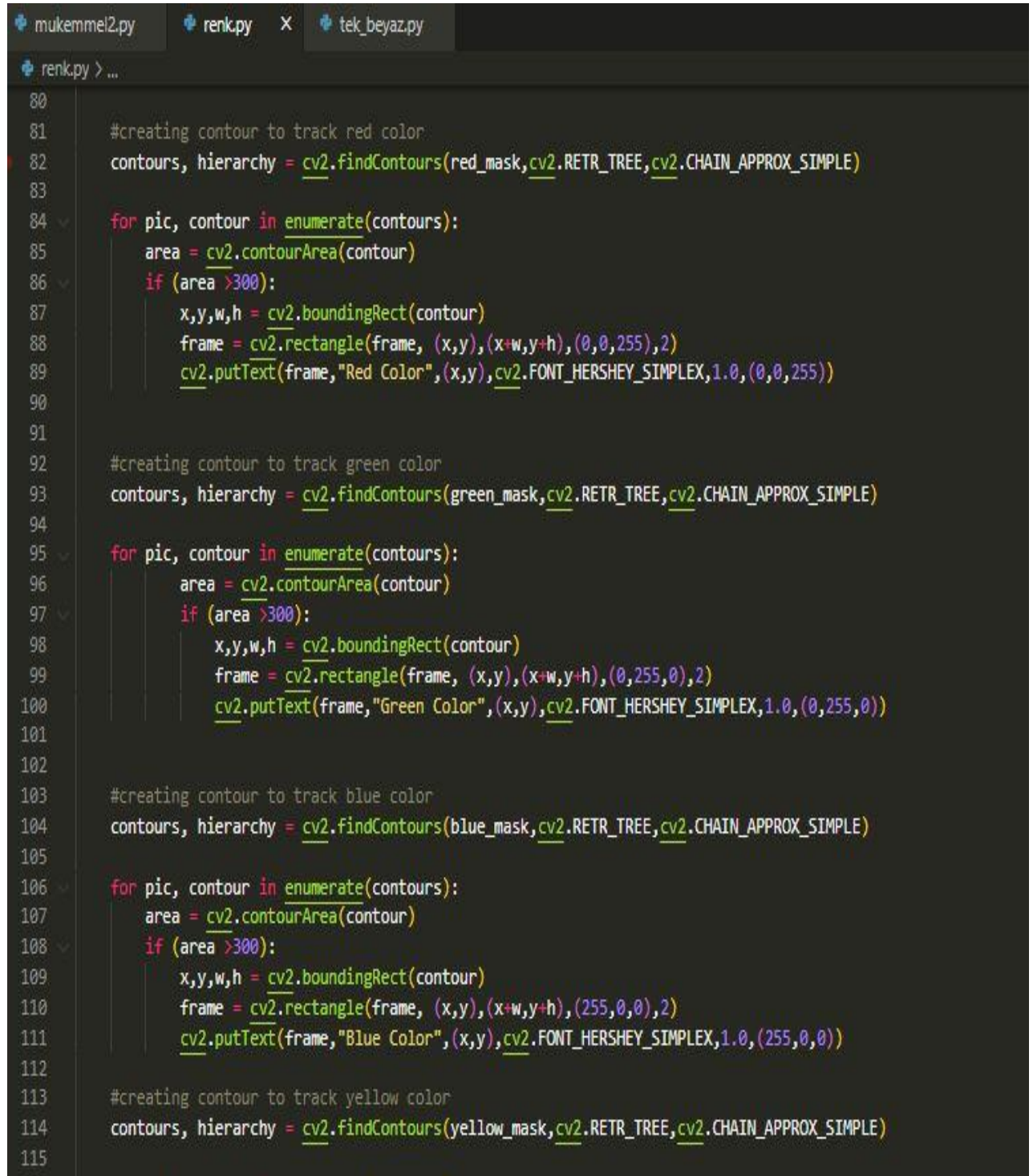
1.5) Araç Renklerini Ayırt Etme

Araçların türünü belirledikten sonra sıra araçların rengini bulmaya geldi. Hedeflediğim renk ile diğer renkler karışmasın diye yeni bir maske ürettim. O maske sayesinde her yer siyah yapıp, sadece belirlediğim renk aralığında olan renkleri görmeyi sağladım.

```
mukemmel2.py x renk.py x tek_beyaz.py
renk.py > ...
11 while(1):
12     frame = webcam.read()
13     hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV) # BGR renk tonundaki videoyu HSV tonuna çevirdim.
14
15     red_lower = np.array([136, 87, 111], np.uint8) # kırmızı renk için düşük değer
16     red_upper = np.array([180, 255, 255], np.uint8) # kırmızı renk için yüksek değer
17     red_mask = cv2.inRange(hsv_frame, red_lower, red_upper) # kırmızı renk için verilen renk aralığı
18     #red = cv2.bitwise_and(frame, frame, mask=red_mask)
19
20     # blue_lower = np.array([94, 80, 2], np.uint8) # mavi renk için düşük değer
21     # blue_upper = np.array([126, 255, 255], np.uint8) # mavi renk için yüksek değer
22     blue_lower = np.array([95, 120, 0], np.uint8) # mavi renk için düşük değer
23     blue_upper = np.array([140, 255, 255], np.uint8) # mavi renk için yüksek değer
24     # blue_lower = np.array([110, 50, 50], np.uint8) # mavi renk için düşük değer
25     # blue_upper = np.array([130, 255, 255], np.uint8) # mavi renk için yüksek değer
26     blue_mask = cv2.inRange(hsv_frame, blue_lower, blue_upper)
27     #blue = cv2.bitwise_and(frame, frame, mask=blue_mask)
28
29     # green_lower = np.array([25, 52, 72], np.uint8) # yeşil renk için düşük değer
30     # green_upper = np.array([102, 255, 255], np.uint8) # yeşil renk için yüksek değer
31     green_lower = np.array([40, 40, 40], np.uint8) # yeşil renk için düşük değer
32     green_upper = np.array([70, 255, 255], np.uint8) # yeşil renk için yüksek değer
33     green_mask = cv2.inRange(hsv_frame, green_lower, green_upper)
34     #green = cv2.bitwise_and(frame, frame, mask=green_mask)
35
36     yellow_lower = np.array([20, 100, 100], np.uint8) # sarı renk için düşük değer
37     yellow_upper = np.array([30, 255, 255], np.uint8) # sarı renk için yüksek değer
38     yellow_mask = cv2.inRange(hsv_frame, yellow_lower, yellow_upper)
39     #yellow = cv2.bitwise_and(frame, frame, mask=yellow_mask)
40
41     black_lower = np.array([0, 0, 0], np.uint8) # siyah renk için düşük değer
42     black_upper = np.array([179, 100, 80], np.uint8) # siyah renk için yüksek değer
43     #black_upper = np.array([180, 255, 30], np.uint8) # siyah renk için yüksek değer
44     black_mask = cv2.inRange(hsv_frame, black_lower, black_upper)
45     #black = cv2.bitwise_and(frame, frame, mask=black_mask)
46
47     white_lower = np.array([0, 0, 205], np.uint8) # beyaz renk için düşük değer
48     #white_lower = np.array([0, 0, 175], np.uint8) # beyaz renk için düşük değer
49     white_upper = np.array([180, 255, 255], np.uint8) # beyaz renk için yüksek değer
50     white_mask = cv2.inRange(hsv_frame, white_lower, white_upper)
51     #white = cv2.bitwise_and(frame, frame, mask=white_mask)
52
```

Şekil 6.1: İstenilen Rengin Renk Aralığını Girme

Renklerin renk aralığı belirlendikten sonra sıra onları göstermede. İstedğimiz rengin ne kadar alandan büyük olması gerektiğini istiyorsak yazıyoruz ($area > 300$). 300'den büyük olanları dikdörtgen içine alıyoruz ve dışına hangi renk olduğunu belirtmek istiyorsak onu yazıyoruz.

The image shows a screenshot of a Python IDE with three tabs: 'mukemmel2.py', 'renk.py', and 'tek_beyaz.py'. The 'renk.py' tab is active, showing a script with line numbers 80 to 115. The code is organized into four sections, each for a different color: red, green, blue, and yellow. Each section starts with a comment '#creating contour to track [color] color', followed by a call to 'cv2.findContours' with parameters (mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE). Then, a 'for' loop with 'enumerate(contours)' iterates over the contours. Inside the loop, 'cv2.contourArea' is used to check if the area is greater than 300. If so, 'cv2.boundingRect' is used to get the bounding box (x, y, w, h), 'cv2.rectangle' is used to draw a rectangle on the frame, and 'cv2.putText' is used to label the rectangle with the color name. The labels are placed at the bottom-left corner of the rectangle. For red, the label is at (0,0,255); for green, at (0,255,0); for blue, at (255,0,0); and for yellow, at (255,0,0).

```
80
81 #creating contour to track red color
82 contours, hierarchy = cv2.findContours(red_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
83
84 for pic, contour in enumerate(contours):
85     area = cv2.contourArea(contour)
86     if (area >300):
87         x,y,w,h = cv2.boundingRect(contour)
88         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
89         cv2.putText(frame,"Red Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,0,255))
90
91
92 #creating contour to track green color
93 contours, hierarchy = cv2.findContours(green_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
94
95 for pic, contour in enumerate(contours):
96     area = cv2.contourArea(contour)
97     if (area >300):
98         x,y,w,h = cv2.boundingRect(contour)
99         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
100        cv2.putText(frame,"Green Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,255,0))
101
102
103 #creating contour to track blue color
104 contours, hierarchy = cv2.findContours(blue_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
105
106 for pic, contour in enumerate(contours):
107     area = cv2.contourArea(contour)
108     if (area >300):
109         x,y,w,h = cv2.boundingRect(contour)
110         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(255,0,0),2)
111         cv2.putText(frame,"Blue Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(255,0,0))
112
113 #creating contour to track yellow color
114 contours, hierarchy = cv2.findContours(yellow_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
115
```

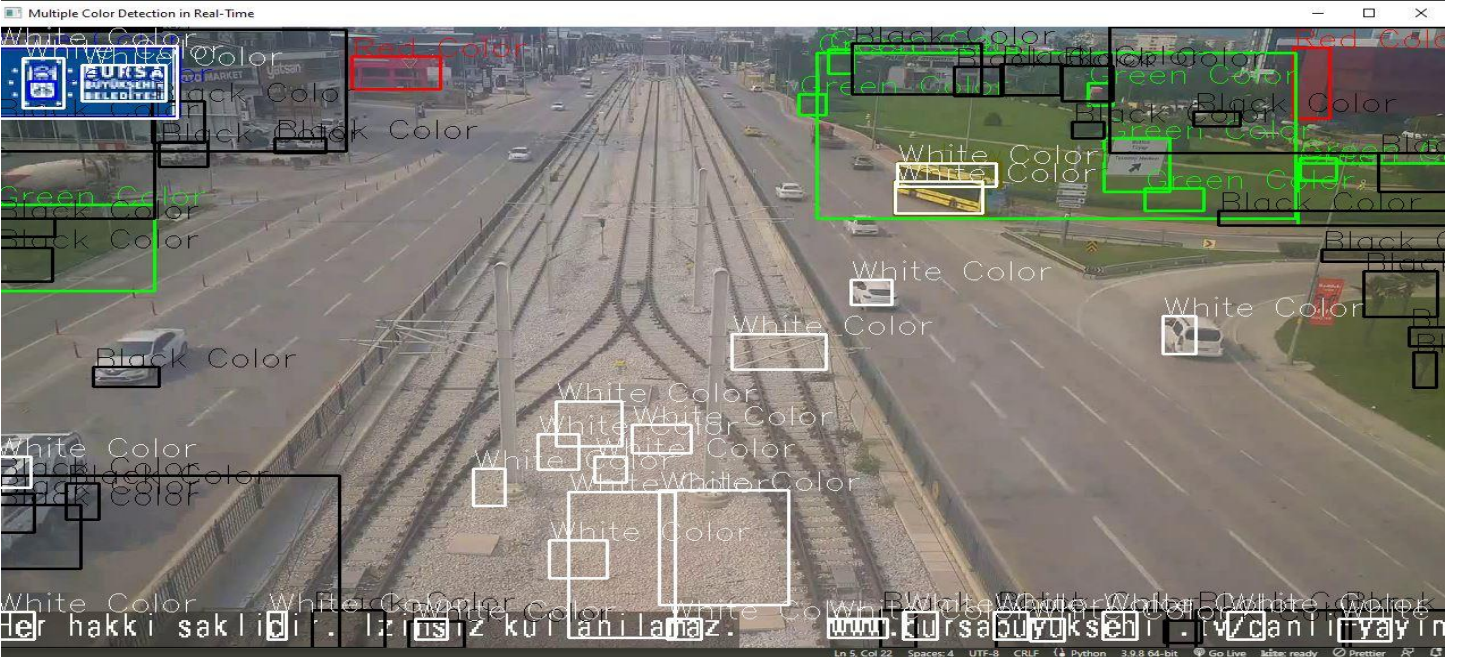
Şekil 6.2: Oluşturulan Maskeler Sayesinde İstenilen Renkteki Aracı Dikdörtgen İçine Alma

```

mukemmel2.py  renk.py  tek_beyaz.py
renk.py > ...
112
113 #creating contour to track yellow color
114 contours, hierarchy = cv2.findContours(yellow_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
115
116 for pic, contour in enumerate(contours):
117     area = cv2.contourArea(contour)
118     if (area > 300):
119         x,y,w,h = cv2.boundingRect(contour)
120         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,100,100),2)
121         cv2.putText(frame,"Yellow Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,100,100))
122
123 #creating contour to track black color
124 contours, hierarchy = cv2.findContours(black_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
125
126 for pic, contour in enumerate(contours):
127     area = cv2.contourArea(contour)
128     if (area > 300):
129         x,y,w,h = cv2.boundingRect(contour)
130         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,0),2)
131         cv2.putText(frame,"Black Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,0,0))
132
133 #creating contour to track white color
134 contours, hierarchy = cv2.findContours(white_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
135
136 for pic, contour in enumerate(contours):
137     area = cv2.contourArea(contour)
138     if (area > 300):
139         x,y,w,h = cv2.boundingRect(contour)
140         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(255, 255, 255),2)
141         cv2.putText(frame,"White Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(255, 255, 255))
142
143
144 cv2.imshow("Multiple Color Detection in Real-Time",frame)
145 if cv2.waitKey(60) & 0xFF == ord('q'):
146     webcam.release()
147     cv2.destroyAllWindows()
148     break
149
150 cv2.destroyAllWindows()
151

```

Şekil 6.2: (devamı)



Şekil 6.3: Renk Kodlarının Uygulanmasıyla Oluşan Görüntü

Araçların hangi saatler arasında yoğun olduğunu bulmak için en çok geçen aracın zaman dilimini bulmak ve hangi indexte olduğunu bulmak için değişkenler tanımladım.

```
87
88 gelen_en_kucuk = gelen_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
89 gelen_en_buyuk = gelen_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
90 gelen_index_kucuk = 0 # gelen araçlardan en küçük değere sahip olanın bulunacağı index
91 gelen_index_buyuk = 0 # gelen araçlardan en yüksek değere sahip olanın bulunacağı index
92 i = 0 # gelen araçlar dizisinin indexini bulmak için tanımlı
93
94 giden_en_kucuk = giden_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
95 giden_en_buyuk = giden_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
96 giden_index_kucuk = 0 # giden araçlardan en küçük değere sahip olanın bulunacağı index
97 giden_index_buyuk = 0 # giden araçlardan en yüksek değere sahip olanın bulunacağı index
98 j = 0 # giden araçlar dizisinin indexini bulmak için tanımlı
99
```

Şekil 7.1: En Çok ve Az Sayıda Geçilen Saatleri Öğrenmek İçin Gerekli Değişkenler

Dizileri teker teker gezerek, dizilerin en büyük ve en küçük değerlerini buluyoruz ve hangi indexte olduğunu buluyoruz.

```
212
213 k=cv2.waitKey(60) & 0xff #esc ye bastığında program kapanır
214 if k == 27 :
215     for n in gelen_arac_sayisi:
216         if gelen_en_kucuk > n:
217             gelen_en_kucuk = n
218             gelen_index_kucuk = i
219         if gelen_en_buyuk < n:
220             gelen_en_buyuk = n
221             gelen_index_buyuk = i
222     i = i + 1
223     #print("Gelen En Büyük Sayı :", gelen_en_buyuk, "\n Gelen En Küçük Sayı :",gelen_en_kucuk)
224     #print("En Büyük index :", gelen_index_buyuk, " :", gelen_arac_sayisi[gelen_index_buyuk] ,"\n En Küçük Index :",gelen_index_kucuk, ":", gelen
225     print("GELEN Araç yoğunluğunun en fazla olduğu aralık :", gecen_zaman[gelen_index_buyuk-1] , " ile " ,gecen_zaman[gelen_index_buyuk] , " aras
226     print("GELEN Araç yoğunluğunun en az olduğu aralık :", gecen_zaman[gelen_index_kucuk-1] , " ile " , gecen_zaman[gelen_index_kucuk] , " arasındadır
227     for m in giden_arac_sayisi:
228         if giden_en_kucuk > m:
229             giden_en_kucuk = m
230             giden_index_kucuk = j
231         if giden_en_buyuk < m:
232             giden_en_buyuk = m
233             giden_index_buyuk = j
234     j = j + 1
235     #print("Giden En Büyük Sayı :", giden_en_buyuk, "\n Giden En Küçük Sayı :",giden_en_kucuk)
236     #print("En Büyük index :", giden_index_buyuk, " :", giden_arac_sayisi[giden_index_buyuk] ,"\n En Küçük Index :",giden_index_kucuk, ":", giden
237     print("GİDEN Araç yoğunluğunun en fazla olduğu aralık :", gecen_zaman[giden_index_buyuk - 1] , " ile " ,gecen_zaman[giden_index_buyuk] , " ara
238     print("GİDEN Araç yoğunluğunun en az olduğu aralık :", gecen_zaman[giden_index_kucuk - 1] , " ile " , gecen_zaman[giden_index_kucuk] , " arasındadır
239     print(gecen_zaman)
240     print(gelen_arac_sayisi)
241     print(giden_arac_sayisi)
242     print(gecen_zaman)
```

Şekil 7.2: Dizi İçindeki En Büyük ve En Küçük Değişkenleri İF Metoduyla Bulma

2. PROJENİN GELİŞTİRME AŞAMASI

Bu son aşamada renk ayrımı için yaptığım projeler ve araç sayma işlemi için yaptığım proje, farklı projeler halinde oluşuyordu. Farklı projeler halinde olduğu için istenilen projeyi çalıştırmak zorunda kalıyordum. Fakat bu aşamada renk ayrımı için yaptığımı, araç sayma projemle birleştirerek hepsini tek projede yaptım. İlk olarak tüm renkleri aynı dosyada, aynı ekranda(frame) oluşturdum. Böyle yaptığımda mesela sarı renk geçtiğinde, diğer renklerinde sayısı artıyordu. Bunu gidermek için her renk için aşağıda fotoğrafını koyduğum şekilde her bir renk için frameler oluşturdum.

```
# resize frame
Kesilmis_Kare= frame
Doldurulmus_Resim=np.zeros((Kesilmis_Kare.shape[0],Kesilmis_Kare.shape[1], 1), np.uint8)

Kesilmis_Kare_tüm= frame
Doldurulmus_Resim_tüm=np.zeros((Kesilmis_Kare_tüm.shape[0],Kesilmis_Kare_tüm.shape[1], 1), np.uint8)

Kesilmis_Kare_red= frame
Doldurulmus_Resim_red=np.zeros((Kesilmis_Kare_red.shape[0],Kesilmis_Kare_red.shape[1], 1), np.uint8)

Kesilmis_Kare_white= frame
Doldurulmus_Resim_white=np.zeros((Kesilmis_Kare_white.shape[0],Kesilmis_Kare_white.shape[1], 1), np.uint8)

Kesilmis_Kare_yellow= frame
Doldurulmus_Resim_yellow=np.zeros((Kesilmis_Kare_yellow.shape[0],Kesilmis_Kare_yellow.shape[1], 1), np.uint8)

Kesilmis_Kare_blue= frame
Doldurulmus_Resim_blue=np.zeros((Kesilmis_Kare_blue.shape[0],Kesilmis_Kare_blue.shape[1], 1), np.uint8)

Kesilmis_Kare_green= frame
Doldurulmus_Resim_green=np.zeros((Kesilmis_Kare_green.shape[0],Kesilmis_Kare_green.shape[1], 1), np.uint8)

kernel=np.ones((5,5), np.uint8)
```

Şekil 8.1: Frameleri Renklerine Göre Ayrı Ayrı Oluşturma

Oluşturduğum her frameleri, dikdörtgen oluşturması için belirlediğimiz renk ayrımı yapan fonksiyonunun içine koyarak, diğer renkler ile karışmasını engelliyoruz ve bu şekilde sayma işleminde de birbirini sayma hatası almadım.

```
204
205 # detect moving anything with loop
206 for cnt in cnts:
207     x,y,w,h=cv2.boundingRect(cnt)
208     if(w>30 and h>30):
209         cv2.rectangle(Sonuc,(x,y),(x+w,y+h),(255,0,0),thickness=2)
210         cv2.rectangle(Doldurulmus_Resim_tüm,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
211
212 #creating contour to track red color
213 contours, hierarchy = cv2.findContours(red_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
214
215 for pic, contour in enumerate(contours):
216     area = cv2.contourArea(contour)
217     if (area >100):
218         x,y,w,h = cv2.boundingRect(contour)
219         frame_red = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
220         cv2.rectangle(Doldurulmus_Resim_red,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
221         cv2.putText(frame_red,"Red Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,0,255))
222
223 #creating contour to track yellow color
224 contours, hierarchy = cv2.findContours(yellow_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
225
226 for pic, contour in enumerate(contours):
227     area = cv2.contourArea(contour)
228     if (area >300):
229         x,y,w,h = cv2.boundingRect(contour)
230         frame_yellow = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,255),2)
231         cv2.rectangle(Doldurulmus_Resim_yellow,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
232         cv2.putText(frame_yellow,"Yellow Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,255,255))
233
234 #creating contour to track white color
235 contours, hierarchy = cv2.findContours(white_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
236
237 for pic, contour in enumerate(contours):
238     area = cv2.contourArea(contour)
239     if (area >350):
240         x,y,w,h = cv2.boundingRect(contour)
241         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(255, 255, 255),2)
242         cv2.rectangle(Doldurulmus_Resim_White,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
243         cv2.putText(frame,"White Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(255, 255, 255))
244
245 contours, hierarchy = cv2.findContours(blue_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
```

Şekil 8.2: Oluşturulan Frame Sahip Olduğu Renk Koduna Göre Dikdörtgene Alma

Her bir renk için oluşturulan frameleri, oranların doğru hesaplanması için de hepsini kendi renklerine gelecek şekilde frameleri yerleştirdim.

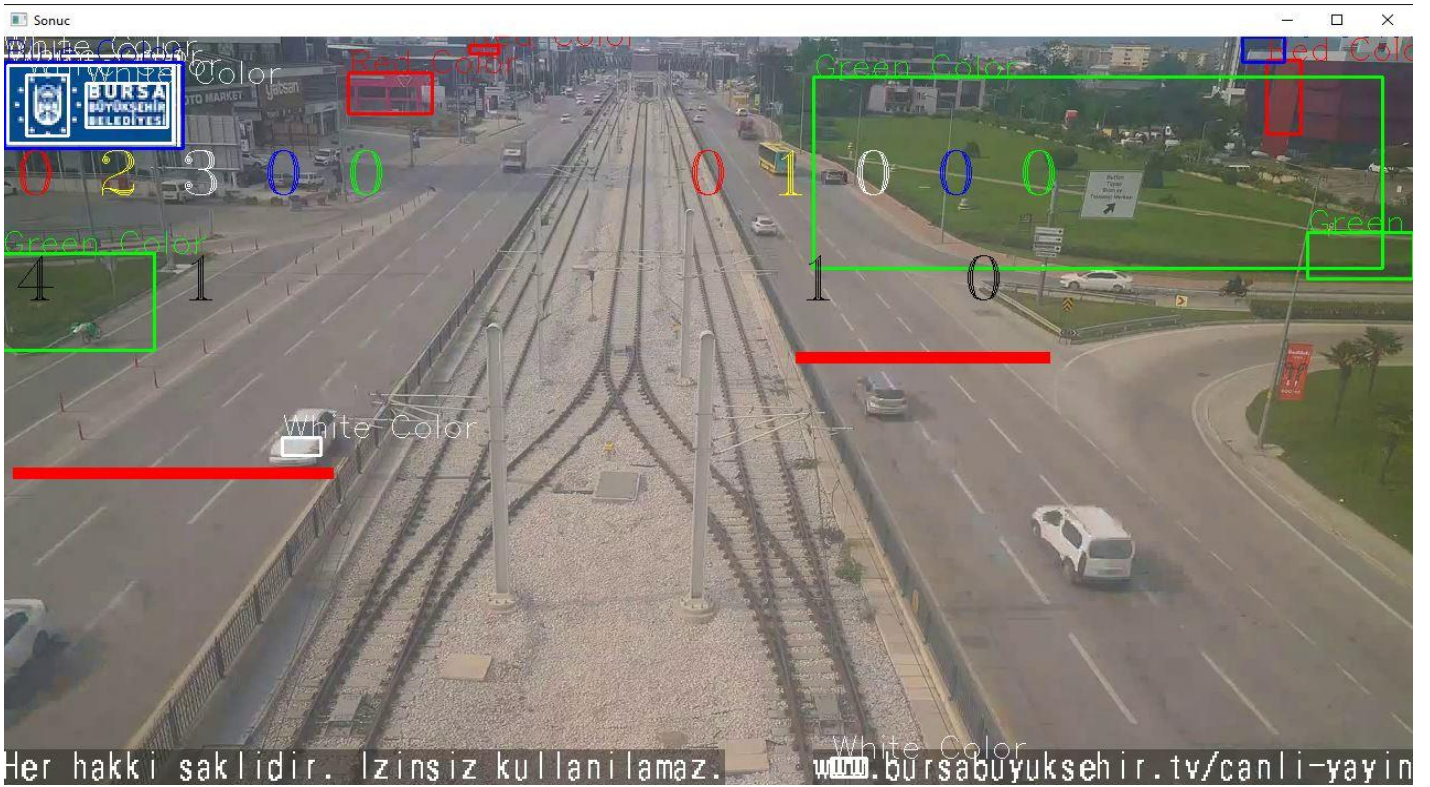
```
331 #KIRMIZI
332 > if (Sensor2.Oran_Kirmizi>=0.5 and Sensor2.Durum==False):...
336 > elif (Sensor2.Oran_Kirmizi>0.1 and Sensor2.Oran_Kirmizi<3 and Sensor2.Durum==True) :...
341 > else :...
344
345
346 > if (Sensor1.Oran_Kirmizi>=0.5 and Sensor1.Durum==False):...
350 > elif (Sensor1.Oran_Kirmizi>0.1 and Sensor1.Durum==True) :...
355 > else :...
358
359
360 #SARI
361 Sensor2_Maske_Sonuc_Sari = cv2.bitwise_or(Doldurulmus_Resim_yellow,Doldurulmus_Resim_yellow,mask=Sensor2.Maske)
362 Sensor2_Beyaz_Piksel_Sayisi_Sari=np.sum(Sensor2_Maske_Sonuc_Sari==255)
363 Sensor2_Oran_Sari=Sensor2_Beyaz_Piksel_Sayisi_Sari/Sensor2.Maskenin_Alani
364
365 Sensor1_Maske_Sonuc_Sari=cv2.bitwise_or(Doldurulmus_Resim_yellow,Doldurulmus_Resim_yellow,mask=Sensor1.Maske)
366 Sensor1_Beyaz_Piksel_Sayisi_Sari=np.sum(Sensor1_Maske_Sonuc_Sari==255)
367 Sensor1_Oran_Sari=Sensor1_Beyaz_Piksel_Sayisi_Sari/Sensor1.Maskenin_Alani
368
369
370 > if (Sensor2.Oran_Sari>=0.5 and Sensor2.Durum==False):...
374 > elif (Sensor2.Oran_Sari>0.3 and Sensor2.Oran_Sari<=4 and Sensor2.Durum==True) :...
379 > else :...
382
383
384 > if (Sensor1.Oran_Sari>=0.5 and Sensor1.Durum==False):...
388 > elif (Sensor1.Oran_Sari>0.3 and Sensor1.Oran_Sari<=2 and Sensor1.Durum==True) :...
393 > else :...
396
397 #BEYAZ
398 Sensor2_Maske_Sonuc_Beyaz = cv2.bitwise_or(Doldurulmus_Resim_White,Doldurulmus_Resim_White,mask=Sensor2.Maske)
399 Sensor2_Beyaz_Piksel_Sayisi_Beyaz=np.sum(Sensor2_Maske_Sonuc_Beyaz==255)
400 Sensor2_Oran_Beyaz=Sensor2_Beyaz_Piksel_Sayisi_Beyaz/Sensor2.Maskenin_Alani
401
402 Sensor1_Maske_Sonuc_Beyaz=cv2.bitwise_or(Doldurulmus_Resim_White,Doldurulmus_Resim_White,mask=Sensor1.Maske)
403 Sensor1_Beyaz_Piksel_Sayisi_Beyaz=np.sum(Sensor1_Maske_Sonuc_Beyaz==255)
404 Sensor1_Oran_Beyaz=Sensor1_Beyaz_Piksel_Sayisi_Beyaz/Sensor1.Maskenin_Alani
405
406
```

Şekil 8.3: Frameleri Renklerine Göre Oranlama

Hangi renk araçtan kaç tane geçtiğini ekranda göstermek için putText fonksiyonunu kullandım. Araçların rengine göre gösterilen sayının rengini de ona göre belirledim.

```
507  
508 cv2.putText(Sonuc, str(Sensor1.Algilanan_Arac_Sayisi), (Sensor1.Koordinat1.x, 250), font, 2, (0, 0, 0))  
509 cv2.putText(Sonuc, str(Sensor1.Algilanan_Uzun_Arac_Sayisi), (Sensor1.Koordinat1.x+150, 250), font, 2, (0, 0, 0))  
510 cv2.putText(Sonuc, str(Sensor2.Algilanan_Arac_Sayisi), (Sensor2.Koordinat1.x, 250), font, 2, (0, 0, 0))  
511 cv2.putText(Sonuc, str(Sensor2.Algilanan_Uzun_Arac_Sayisi), (Sensor2.Koordinat1.x+150, 250), font, 2, (0, 0, 0))  
512  
513 cv2.putText(Sonuc, str(Sensor1.Kirmizi_Arac), (Sensor1.Koordinat1.x-100, 150), font, 2, (0, 0, 255))  
514 cv2.putText(Sonuc, str(Sensor2.Kirmizi_Arac), (Sensor2.Koordinat1.x, 150), font, 2, (0, 0, 255))  
515 cv2.putText(Sonuc, str(Sensor1.Sari_Arac), (Sensor1.Koordinat1.x-25, 150), font, 2, (0, 255, 255))  
516 cv2.putText(Sonuc, str(Sensor2.Sari_Arac), (Sensor2.Koordinat1.x+75, 150), font, 2, (0, 255, 255))  
517 cv2.putText(Sonuc, str(Sensor1.Beyaz_Arac), (Sensor1.Koordinat1.x+50, 150), font, 2, (255, 255, 255))  
518 cv2.putText(Sonuc, str(Sensor2.Beyaz_Arac), (Sensor2.Koordinat1.x+150, 150), font, 2, (255, 255, 255))  
519 cv2.putText(Sonuc, str(Sensor1.Mavi_Arac), (Sensor1.Koordinat1.x+125, 150), font, 2, (255, 0, 0))  
520 cv2.putText(Sonuc, str(Sensor2.Mavi_Arac), (Sensor2.Koordinat1.x+225, 150), font, 2, (255, 0, 0))  
521 cv2.putText(Sonuc, str(Sensor1.Yesil_Arac), (Sensor1.Koordinat1.x+200, 150), font, 2, (0, 255, 0))  
522 cv2.putText(Sonuc, str(Sensor2.Yesil_Arac), (Sensor2.Koordinat1.x+300, 150), font, 2, (0, 255, 0))  
523
```

Şekil 8.4: putText Fonksiyonun Kullanımı



Şekil 8.5: Kodun Çıktısı

3. SONUÇLAR

İlk başta bu projeyi yapmak için Python bilmem gerektiğini biliyordum. Fakat daha önce görmediğim için benim için riskli bir projeydi. Fakat projemi yaparken genellikle araştırarak bulmam gerektiğini anladım ve birazda yazılım bilgim olduğundan Python dilini kolay anladım ve kullanımının kolay ve çok yönlü olduğunu fark ettim. Sonuç olarak projemde araçları algılama ve sayma işlemini, canlı olarak Mobese kamerasından trafik akışını çekme, renklerine göre ayırt etme, verileri kaydetme ve grafiksel olarak gösterme işlemlerini başarılı şekilde gerçekleştirdim. Renk ayrımı için kod aralığı çok geniş olduğu için, çekilen Mobese kaydının kalitesi olsun, havanın durumu olsun çok etkili olduğu için onlarda küçük sapmalar yaşanabiliyor. Bu proje için yazdığım tüm kodları ve oluşan çıktıları aşağıdaki sayfalarda yer almaktadır.

birlestir_copy.py > ...

```
1 import cv2
2 import numpy as np
3 from datetime import datetime, timedelta
4 import matplotlib.pyplot as plt
5 import xlswriter
6
7 planWorkBook = xlswriter.Workbook('veri.xlsx') #kaydedilecek dosyanın adı
8 planSheet = planWorkBook.add_worksheet("sayfaAdi")
9 planSheet.write("A1","GEÇEN ZAMAN") # 1. sütunun adı
10 planSheet.write("B1","GELEN ARAC SAYISI") # 2. sütunun adı
11 planSheet.write("C1","GELEN UZUN ARAC SAYISI") # 2. sütunun adı
12 planSheet.write("D1","GİDEN ARAC SAYISI") # 3. sütunun adı
13 planSheet.write("E1","GİDEN UZUN ARAC SAYISI") # 2. sütunun adı
14 planSheet.write("F1","GELEN BEYAZ ARAC SAYISI")
15 planSheet.write("G1","GİDEN BEYAZ ARAC SAYISI")
16 planSheet.write("H1","GELEN KIRMIZI ARAC SAYISI")
17 planSheet.write("I1","GİDEN KIRMIZI ARAC SAYISI")
18 planSheet.write("J1","GELEN MAVİ ARAC SAYISI")
19 planSheet.write("K1","GİDEN MAVİ ARAC SAYISI")
20 planSheet.write("L1","GELEN SARI ARAC SAYISI")
21 planSheet.write("M1","GİDEN SARI ARAC SAYISI")
22 planSheet.write("N1","GELEN YEŞİL ARAC SAYISI")
23 planSheet.write("O1","GİDEN YEŞİL ARAC SAYISI")
24
25
26
27 def rescale(frame, scale=1):
28     width = int(frame.shape[1] * scale)
29     height = int(frame.shape[0] * scale)
30
31     if (width<=0 or height<=0):
32         width = frame.shape[1]
33         height = frame.shape[0]
34
35     boyut = (width, height)
36     return cv2.resize(frame, boyut, interpolation=cv2.INTER_AREA)
37
38 class Koordinat:
39     def __init__(self,x,y):
40         self.x=x
41         self.y=y
42
43 class Sensor:
44     def __init__(self,Koordinat1,Koordinat2,Kare_Genislik,Kare_Yukseklk):
45         self.Koordinat1 = Koordinat1
46         self.Koordinat2 = Koordinat2
47         self.Kare_Genislik = Kare_Genislik
48         self.Kare_Yukseklk = Kare_Yukseklk
49         self.MaskeninAlan = abs(self.Koordinat2.x-self.Koordinat1.x)
50         self.Maske=np.zeros((Kare_Genislik,Kare_Yukseklk,1),np.uint8)*abs(self.Koordinat2.y-self.Koordinat1.y)
51         cv2.rectangle(self.Maske,(self.Koordinat1.x,self.Koordinat1.y),(self.Koordinat2.x,self.Koordinat2.y),(255),thickness=cv2.FILLED)
52         self.Durum=False
53         self.Algilanan_Arac_Sayisi=0
54         self.Algilanan_Uzun_Arac_Sayisi = 0
55         self.Sari_Arac = 0
56         self.Kirmizi_Arac = 0
57         self.Beyaz_Arac = 0
58         self.Mavi_Arac = 0
59         self.Yesil_Arac = 0
60
61
62 gelen_arac_sayisi = []
63 giden_arac_sayisi = []
64 gelen_uzun_arac_sayisi = []
65 giden_uzun_arac_sayisi = []
66 gelen_kirmizi_arac_sayisi = []
67 giden_kirmizi_arac_sayisi = []
68 gelen_beyaz_arac_sayisi = []
69 giden_beyaz_arac_sayisi = []
70 gelen_mavi_arac_sayisi = []
71 giden_mavi_arac_sayisi = []
72 gelen_sari_arac_sayisi = []
73 giden_sari_arac_sayisi = []
74 gelen_yesil_arac_sayisi = []
75 giden_yesil_arac_sayisi = []
```



```

75 giden_yesil_arac_sayisi = []
76 gecen_zaman = []
77 gelen_arac_sayisi.append(0)
78 giden_arac_sayisi.append(0)
79 gelen_uzun_arac_sayisi.append(0)
80 giden_uzun_arac_sayisi.append(0)
81 gelen_kirmizi_arac_sayisi.append(0)
82 giden_kirmizi_arac_sayisi.append(0)
83 gelen_beyaz_arac_sayisi.append(0)
84 giden_beyaz_arac_sayisi.append(0)
85 gelen_mavi_arac_sayisi.append(0)
86 giden_mavi_arac_sayisi.append(0)
87 gelen_sari_arac_sayisi.append(0)
88 giden_sari_arac_sayisi.append(0)
89 gelen_yesil_arac_sayisi.append(0)
90 giden_yesil_arac_sayisi.append(0)
91
92 aralik = 1
93 aralikT0saniye = aralik*60;
94 now = datetime.now()
95 gecen_zaman.append(now.strftime("%H:%M"))
96 video_bölme_bitis_zamanı = now + timedelta(seconds=aralikT0saniye)
97
98
99 video = "./video2.avi"
100 Video_Okuyucu=cv2.VideoCapture(video )
101 ret,Kare=Video_Okuyucu.read()
102 Kesilmis_Kare= Kare
103 #print(Kesilmis_Kare.shape[0])
104 #print(Kesilmis_Kare.shape[1])
105 fgbg=cv2.createBackgroundSubtractorMOG2()
106
107 Sensor1 = Sensor(
108     Koordinat(720, Kesilmis_Kare.shape[1] - 980),
109     Koordinat(950, Kesilmis_Kare.shape[1] - 970),
110     Kesilmis_Kare.shape[0],
111     Kesilmis_Kare.shape[1])
112
113 Sensor2 = Sensor(
114     Koordinat(10, 420),
115     Koordinat(300, 410),
116     Kesilmis_Kare.shape[0],
117     Kesilmis_Kare.shape[1])
118
119 font=cv2.FONT_HERSHEY_TRIPLEX
120
121 kernal = np.ones((5,5), "uint8")
122 fgbg=cv2.createBackgroundSubtractorMOG2()
123
124 gelen_en_kucuk = gelen_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
125 gelen_en_buyuk =gelen_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
126 gelen_index_kucuk = 0 # gelen araçlardan en küçük değere sahip olanın bulunacağı index
127 gelen_index_buyuk = 0 # gelen araçlardan en yüksek değere sahip olanın bulunacağı index
128 i = 0 # gelen araçlar dizisinin indexini bulmak için tanımlı
129
130 giden_en_kucuk = giden_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
131 giden_en_buyuk = giden_arac_sayisi[0] # dizinin ilk elemanını atadım. Fonksiyonla kontrol edileceğinden önemi pek yok
132 giden_index_kucuk = 0 # giden araçlardan en küçük değere sahip olanın bulunacağı index
133 giden_index_buyuk = 0 # giden araçlardan en yüksek değere sahip olanın bulunacağı index
134 j = 0 # giden araçlar dizisinin indexini bulmak için tanımlı
135
136 while (1):
137     _, frame = Video_Okuyucu.read()
138     frame = rescale(frame)
139     hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
140
141     red_lower = np.array([136,87,111],np.uint8)
142     red_upper = np.array([180, 255, 255],np.uint8)
143     red_mask = cv2.inRange(hsv_frame, red_lower, red_upper)
144
145     red_mask = cv2.dilate(red_mask,kernal)
146     res_red = cv2.bitwise_and(frame, frame, mask= red_mask)
147
148     yellow_lower = np.array([20, 100, 100],np.uint8)
149     yellow_upper = np.array([30, 255, 255],np.uint8)

```

```

148 yellow_lower = np.array([20, 100, 100], np.uint8)
149 yellow_upper = np.array([30, 255, 255], np.uint8)
150 yellow_mask = cv2.inRange(hsv_frame, yellow_lower, yellow_upper)
151
152 yellow_mask = cv2.dilate(yellow_mask, kernal)
153 res_yellow = cv2.bitwise_and(frame, frame, mask=yellow_mask)
154
155 white_lower = np.array([40, 0, 193], np.uint8)
156 white_upper = np.array([180, 255, 255], np.uint8)
157 white_mask = cv2.inRange(hsv_frame, white_lower, white_upper)
158
159 white_mask = cv2.dilate(white_mask, kernal)
160 res_white = cv2.bitwise_and(frame, frame, mask=white_mask)
161
162 blue_lower = np.array([94, 80, 2], np.uint8)
163 blue_upper = np.array([140, 255, 255], np.uint8)
164 blue_mask = cv2.inRange(hsv_frame, blue_lower, blue_upper)
165
166 blue_mask = cv2.dilate(blue_mask, kernal)
167 res_blue = cv2.bitwise_and(frame, frame, mask=blue_mask)
168
169 green_lower = np.array([40, 55, 80], np.uint8)
170 green_upper = np.array([70, 255, 255], np.uint8)
171 green_mask = cv2.inRange(hsv_frame, green_lower, green_upper)
172
173 green_mask = cv2.dilate(green_mask, kernal)
174 res_green = cv2.bitwise_and(frame, frame, mask=green_mask)
175
176 # resize frame
177 Kesilmis_Kare = frame
178 Doldurulmus_Resim = np.zeros((Kesilmis_Kare.shape[0], Kesilmis_Kare.shape[1], 1), np.uint8)
179
180 Kesilmis_Kare_tüm = frame
181 Doldurulmus_Resim_tüm = np.zeros((Kesilmis_Kare_tüm.shape[0], Kesilmis_Kare_tüm.shape[1], 1), np.uint8)
182
183 Kesilmis_Kare_red = frame
184 Doldurulmus_Resim_red = np.zeros((Kesilmis_Kare_red.shape[0], Kesilmis_Kare_red.shape[1], 1), np.uint8)
185
186 Kesilmis_Kare_white = frame
187 Doldurulmus_Resim_white = np.zeros((Kesilmis_Kare_white.shape[0], Kesilmis_Kare_white.shape[1], 1), np.uint8)
188
189 Kesilmis_Kare_yellow = frame
190 Doldurulmus_Resim_yellow = np.zeros((Kesilmis_Kare_yellow.shape[0], Kesilmis_Kare_yellow.shape[1], 1), np.uint8)
191
192 Kesilmis_Kare_blue = frame
193 Doldurulmus_Resim_blue = np.zeros((Kesilmis_Kare_blue.shape[0], Kesilmis_Kare_blue.shape[1], 1), np.uint8)
194
195 Kesilmis_Kare_green = frame
196 Doldurulmus_Resim_green = np.zeros((Kesilmis_Kare_green.shape[0], Kesilmis_Kare_green.shape[1], 1), np.uint8)
197
198 kernel = np.ones((5, 5), np.uint8)
199 Arka_Plani_Silinmis_Kare = fgbg.apply(Kesilmis_Kare_tüm)
200 Arka_Plani_Silinmis_Kare = cv2.morphologyEx(Arka_Plani_Silinmis_Kare, cv2.MORPH_OPEN, kernel)
201 ret, Arka_Plani_Silinmis_Kare = cv2.threshold(Arka_Plani_Silinmis_Kare, 127, 255, cv2.THRESH_BINARY)
202 cnts, _ = cv2.findContours(Arka_Plani_Silinmis_Kare, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
203 #Sonuc = Kesilmis_Kare.copy()
204
205 # detect moving anything with loop
206 for cnt in cnts:
207     x, y, w, h = cv2.boundingRect(cnt)
208     if (w > 30 and h > 30):
209         cv2.rectangle(Sonuc, (x, y), (x+w, y+h), (255, 0, 0), thickness=2)
210         cv2.rectangle(Doldurulmus_Resim_tüm, (x, y), (x+w, y+h), (255), thickness=cv2.FILLED)
211
212 #creating contour to track red color
213 contours, hierarchy = cv2.findContours(res_mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
214
215 for pic, contour in enumerate(contours):
216     area = cv2.contourArea(contour)
217     if (area > 100):
218         x, y, w, h = cv2.boundingRect(contour)
219         frame_red = cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255), 2)
220         cv2.rectangle(Doldurulmus_Resim_red, (x, y), (x+w, y+h), (255), thickness=cv2.FILLED)
221         cv2.putText(frame_red, "Red Color", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 0, 255))
222
223 #creating contour to track yellow color

```



```

219     frame_red = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
220     cv2.rectangle(Doldurulumus_Resim_red,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
221     cv2.putText(frame_red,"Red Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,0,255))
222
223 #creating contour to track yellow color
224 contours, hierarchy = cv2.findContours(yellow_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
225
226 for pic, contour in enumerate(contours):
227     area = cv2.contourArea(contour)
228     if (area >300):
229         x,y,w,h = cv2.boundingRect(contour)
230         frame_yellow = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,255),2)
231         cv2.rectangle(Doldurulumus_Resim_yellow,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
232         cv2.putText(frame_yellow,"Yellow Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,255,255))
233
234 #creating contour to track white color
235 contours, hierarchy = cv2.findContours(white_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
236
237 for pic, contour in enumerate(contours):
238     area = cv2.contourArea(contour)
239     if (area >350):
240         x,y,w,h = cv2.boundingRect(contour)
241         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(255, 255, 255),2)
242         cv2.rectangle(Doldurulumus_Resim_White,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
243         cv2.putText(frame,"White Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(255, 255, 255))
244
245 contours, hierarchy = cv2.findContours(blue_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
246
247 for pic, contour in enumerate(contours):
248     area = cv2.contourArea(contour)
249     if (area >200):
250         x,y,w,h = cv2.boundingRect(contour)
251         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(255,0,0),2)
252         cv2.rectangle(Doldurulumus_Resim_blue,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
253         cv2.putText(frame,"Blue Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(255,0,0))
254
255 contours, hierarchy = cv2.findContours(green_mask,cv2.RETR_TREE,cv2.CHAIN_APPROX_SIMPLE)
256
257 for pic, contour in enumerate(contours):
258     area = cv2.contourArea(contour)
259     if (area >400):
260         x,y,w,h = cv2.boundingRect(contour)
261         frame = cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
262         cv2.rectangle(Doldurulumus_Resim_green,(x,y),(x+w,y+h),(255),thickness=cv2.FILLED)
263         cv2.putText(frame,"Green Color",(x,y),cv2.FONT_HERSHEY_SIMPLEX,1.0,(0,255,0))
264
265
266 # make morphology for frame
267 Arka_Plani_Silinmis_Kare=fgbg.apply(Kesilmis_Kare)
268 Arka_Plani_Silinmis_Kare=cv2.morphologyEx(Arka_Plani_Silinmis_Kare,cv2.MORPH_OPEN,kernal)
269 ret,Arka_Plani_Silinmis_Kare=cv2.threshold(Arka_Plani_Silinmis_Kare,127,255,cv2.THRESH_BINARY)
270
271 # detect moving anything
272 cnts,_=cv2.findContours(Arka_Plani_Silinmis_Kare,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
273 Sonuc=Kesilmis_Kare.copy()
274
275 Sensor1_Maske_Sonuc=cv2.bitwise_or(Doldurulumus_Resim_tüm,Doldurulumus_Resim_tüm,mask=Sensor1.Maske)
276 Sensor1_Beyaz_Piksel_Sayisi=np.sum(Sensor1_Maske_Sonuc==255)
277 Sensor1_Oran=Sensor1_Beyaz_Piksel_Sayisi/Sensor1.Maskenin_Alanı
278
279 Sensor2_Maske_Sonuc = cv2.bitwise_or(Doldurulumus_Resim_tüm,Doldurulumus_Resim_tüm,mask=Sensor2.Maske)
280 Sensor2_Beyaz_Piksel_Sayisi=np.sum(Sensor2_Maske_Sonuc==255)
281 Sensor2_Oran=Sensor2_Beyaz_Piksel_Sayisi/Sensor2.Maskenin_Alanı
282
283 if (Sensor1_Oran>=0.5 and Sensor1.Durum==False):
284     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
285     (0,255, 0, ), thickness=cv2.FILLED)
286     Sensor1.Durum = True
287 elif (Sensor1_Oran>1 and Sensor1_Oran<=4 and Sensor1.Durum==True) :
288     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
289     (0, 0,255), thickness=cv2.FILLED)
290     Sensor1.Durum = False
291     Sensor1.Algilanan_Arac_Sayisi+=1
292 elif (Sensor1_Oran>5.5 and Sensor1_Oran<=7.5 and Sensor1.Durum==True) :
293     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
294     (0, 255,255), thickness=cv2.FILLED)

```

```

290 Sensor1.Durum = False
291 Sensor1.Algilanan_Arac_Sayisi+=1
292 elif (Sensor1.Oran>5.5 and Sensor1.Oran<=7.5 and Sensor1.Durum==True) :
293     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
294                   (0, 255,255), thickness=cv2.FILLED)
295     Sensor1.Durum = False
296     Sensor1.Algilanan_Uzun_Arac_Sayisi += 1
297 else :
298     cv2.rectangle(Sonuc, (Sensor1.Koordinat1.x, Sensor1.Koordinat1.y), (Sensor1.Koordinat2.x, Sensor1.Koordinat2.y),
299                   (0, 0, 255), thickness=cv2.FILLED)
300
301
302 if (Sensor2.Oran>=0.5 and Sensor2.Durum==False):
303     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
304                   (0,255, 0),, thickness=cv2.FILLED)
305     Sensor2.Durum = True
306 elif (Sensor2.Oran>1 and Sensor2.Oran<=4 and Sensor2.Durum==True) :
307     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
308                   (0, 0,255), thickness=cv2.FILLED)
309     Sensor2.Durum = False
310     Sensor2.Algilanan_Arac_Sayisi+=1
311 elif (Sensor2.Oran>5.45 and Sensor2.Oran<=7.2 and Sensor2.Durum==True) :
312     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
313                   (0, 255,255), thickness=cv2.FILLED)
314     Sensor2.Durum = False
315     Sensor2.Algilanan_Uzun_Arac_Sayisi += 1
316 else :
317     cv2.rectangle(Sonuc, (Sensor2.Koordinat1.x, Sensor2.Koordinat1.y), (Sensor2.Koordinat2.x, Sensor2.Koordinat2.y),
318                   (0, 0, 255), thickness=cv2.FILLED)
319
320 # detect whether there is car via bitwise_and
321
322 Sensor2_Maske_Sonuc_Kirmizi = cv2.bitwise_or(Doldurulmus_Resim_red,Doldurulmus_Resim_red,mask=Sensor2.Maske)
323 Sensor2_Beyaz_Piksel_Sayisi_Kirmizi=np.sum(Sensor2_Maske_Sonuc_Kirmizi==255)
324 Sensor2.Oran_Kirmizi=Sensor2_Beyaz_Piksel_Sayisi_Kirmizi/Sensor2.Maskenin_Alani
325
326 Sensor1_Maske_Sonuc_Kirmizi=cv2.bitwise_or(Doldurulmus_Resim_red,Doldurulmus_Resim_red,mask=Sensor1.Maske)
327 Sensor1_Beyaz_Piksel_Sayisi_Kirmizi=np.sum(Sensor1_Maske_Sonuc_Kirmizi==255)
328 Sensor1.Oran_Kirmizi=Sensor1_Beyaz_Piksel_Sayisi_Kirmizi/Sensor1.Maskenin_Alani
329
330
331 #KIRMIZI
332 > if (Sensor2.Oran_Kirmizi>=0.5 and Sensor2.Durum==False):...
336 > elif (Sensor2.Oran_Kirmizi>0.1 and Sensor2.Oran_Kirmizi<3 and Sensor2.Durum==True) : ...
341 > else : ...
344
345
346 > if (Sensor1.Oran_Kirmizi>=0.5 and Sensor1.Durum==False):...
350 > elif (Sensor1.Oran_Kirmizi>0.1 and Sensor1.Durum==True) : ...
355 > else : ...
358
359
360 #SARI
361 Sensor2_Maske_Sonuc_Sari = cv2.bitwise_or(Doldurulmus_Resim_yellow,Doldurulmus_Resim_yellow,mask=Sensor2.Maske)
362 Sensor2_Beyaz_Piksel_Sayisi_Sari=np.sum(Sensor2_Maske_Sonuc_Sari==255)
363 Sensor2.Oran_Sari=Sensor2_Beyaz_Piksel_Sayisi_Sari/Sensor2.Maskenin_Alani
364
365 Sensor1_Maske_Sonuc_Sari=cv2.bitwise_or(Doldurulmus_Resim_yellow,Doldurulmus_Resim_yellow,mask=Sensor1.Maske)
366 Sensor1_Beyaz_Piksel_Sayisi_Sari=np.sum(Sensor1_Maske_Sonuc_Sari==255)
367 Sensor1.Oran_Sari=Sensor1_Beyaz_Piksel_Sayisi_Sari/Sensor1.Maskenin_Alani
368
369
370 > if (Sensor2.Oran_Sari>=0.5 and Sensor2.Durum==False):...
374 > elif (Sensor2.Oran_Sari>0.3 and Sensor2.Oran_Sari<=4 and Sensor2.Durum==True) : ...
379 > else : ...
382
383
384 > if (Sensor1.Oran_Sari>=0.5 and Sensor1.Durum==False):...
388 > elif (Sensor1.Oran_Sari>0.3 and Sensor1.Oran_Sari<=2 and Sensor1.Durum==True) : ...
393 > else : ...
396
397 #BEYAZ

```


birlestir_copy.py > ...

```
393 > else : ...
396
397 #BEYAZ
398 Sensor2_Maske_Sonuc_Beyaz = cv2.bitwise_or(Doldurulmus_Resim_White,Doldurulmus_Resim_White,mask=Sensor2.Maske)
399 Sensor2_Beyaz_Piksel_Sayisi_Beyaz=np.sum(Sensor2_Maske_Sonuc_Beyaz==255)
400 Sensor2_Oran_Beyaz=Sensor2_Beyaz_Piksel_Sayisi_Beyaz/Sensor2.Maskenin_Alani
401
402 Sensor1_Maske_Sonuc_Beyaz=cv2.bitwise_or(Doldurulmus_Resim_White,Doldurulmus_Resim_White,mask=Sensor1.Maske)
403 Sensor1_Beyaz_Piksel_Sayisi_Beyaz=np.sum(Sensor1_Maske_Sonuc_Beyaz==255)
404 Sensor1_Oran_Beyaz=Sensor1_Beyaz_Piksel_Sayisi_Beyaz/Sensor1.Maskenin_Alani
405
406
407 > if (Sensor2_Oran_Beyaz>=0.5 and Sensor2.Durum==False): ...
411 > elif (Sensor2_Oran_Beyaz>0.7 and Sensor2_Oran_Beyaz<=4.5 and Sensor2.Durum==True) : ...
416 > else : ...
419
420
421 > if (Sensor1_Oran_Beyaz>=0.5 and Sensor1.Durum==False): ...
425 > elif (Sensor1_Oran_Beyaz>1 and Sensor1_Oran_Beyaz<=3 and Sensor1.Durum==True) : ...
430 > else : ...
433
434 #MAVI
435 Sensor2_Maske_Sonuc_blue = cv2.bitwise_or(Doldurulmus_Resim_blue,Doldurulmus_Resim_blue,mask=Sensor2.Maske)
436 Sensor2_Beyaz_Piksel_Sayisi_blue=np.sum(Sensor2_Maske_Sonuc_blue==255)
437 Sensor2_Oran_Blue=Sensor2_Beyaz_Piksel_Sayisi_blue/Sensor2.Maskenin_Alani
438
439 Sensor1_Maske_Sonuc_blue=cv2.bitwise_or(Doldurulmus_Resim_blue,Doldurulmus_Resim_blue,mask=Sensor1.Maske)
440 Sensor1_Beyaz_Piksel_Sayisi_blue=np.sum(Sensor1_Maske_Sonuc_blue==255)
441 Sensor1_Oran_Blue=Sensor1_Beyaz_Piksel_Sayisi_blue/Sensor1.Maskenin_Alani
442
443
444 > if (Sensor2_Oran_Blue>=0.5 and Sensor2.Durum==False): ...
448 > elif (Sensor2_Oran_Blue>0.1 and Sensor2_Oran_Blue<=4 and Sensor2.Durum==True) : ...
453 > else : ...
456
457
458 > if (Sensor1_Oran_Blue>=0.5 and Sensor1.Durum==False): ...
462 > elif (Sensor1_Oran_Blue>0.1 and Sensor1_Oran_Blue<=4 and Sensor1.Durum==True) : ...
467 > else : ...
470
471
472 Sensor2_Maske_Sonuc_green = cv2.bitwise_or(Doldurulmus_Resim_green,Doldurulmus_Resim_green,mask=Sensor2.Maske)
473 Sensor2_Beyaz_Piksel_Sayisi_green=np.sum(Sensor2_Maske_Sonuc_green==255)
474 Sensor2_Oran_Green=Sensor2_Beyaz_Piksel_Sayisi_green/Sensor2.Maskenin_Alani
475
476 Sensor1_Maske_Sonuc_green=cv2.bitwise_or(Doldurulmus_Resim_green,Doldurulmus_Resim_green,mask=Sensor1.Maske)
477 Sensor1_Beyaz_Piksel_Sayisi_green=np.sum(Sensor1_Maske_Sonuc_green==255)
478 Sensor1_Oran_Green=Sensor1_Beyaz_Piksel_Sayisi_green/Sensor1.Maskenin_Alani
479
480 > if (Sensor2_Oran_Green>=0.5 and Sensor2.Durum==False): ...
484 > elif (Sensor2_Oran_Green>0.3 and Sensor2_Oran_Green<=3 and Sensor2.Durum==True) : ...
489 > else : ...
492
493
494 > if (Sensor1_Oran_Green>=0.5 and Sensor1.Durum==False): ...
498 > elif (Sensor1_Oran_Green>0.3 and Sensor1_Oran_Green<=3 and Sensor1.Durum==True) : ...
503 > else : ...
506
507
508 cv2.putText(Sonuc,str(Sensor1.Algilanan_Arac_Sayisi),(Sensor1.Koordinat1.x,250),font,2,(0,0,0))
509 cv2.putText(Sonuc,str(Sensor1.Algilanan_Uzun_Arac_Sayisi),(Sensor1.Koordinat1.x+150,250),font,2,(0,0,0))
510 cv2.putText(Sonuc,str(Sensor2.Algilanan_Arac_Sayisi),(Sensor2.Koordinat1.x,250),font,2,(0,0,0))
511 cv2.putText(Sonuc,str(Sensor2.Algilanan_Uzun_Arac_Sayisi),(Sensor2.Koordinat1.x+150,250),font,2,(0,0,0))
512
513 cv2.putText(Sonuc,str(Sensor1.Kirmizi_Arac),(Sensor1.Koordinat1.x-100,150),font,2,(0,0,255))
514 cv2.putText(Sonuc,str(Sensor2.Kirmizi_Arac),(Sensor2.Koordinat1.x,150),font,2,(0,0,255))
515 cv2.putText(Sonuc,str(Sensor1.Sari_Arac),(Sensor1.Koordinat1.x-25,150),font,2,(0,255,255))
516 cv2.putText(Sonuc,str(Sensor2.Sari_Arac),(Sensor2.Koordinat1.x+75,150),font,2,(0,255,255))
517 cv2.putText(Sonuc,str(Sensor1.Beyaz_Arac),(Sensor1.Koordinat1.x+50,150),font,2,(255,255,255))
518 cv2.putText(Sonuc,str(Sensor2.Beyaz_Arac),(Sensor2.Koordinat1.x+150,150),font,2,(255,255,255))
519 cv2.putText(Sonuc,str(Sensor1.Mavi_Arac),(Sensor1.Koordinat1.x+125,150),font,2,(255,0,0))
520 cv2.putText(Sonuc,str(Sensor2.Mavi_Arac),(Sensor2.Koordinat1.x+225,150),font,2,(255,0,0))
521 cv2.putText(Sonuc,str(Sensor1.Yesil_Arac),(Sensor1.Koordinat1.x+200,150),font,2,(0,255,0))
522 cv2.putText(Sonuc,str(Sensor2.Yesil_Arac),(Sensor2.Koordinat1.x+300,150),font,2,(0,255,0))
523
```


birlestir_copy.py > ...

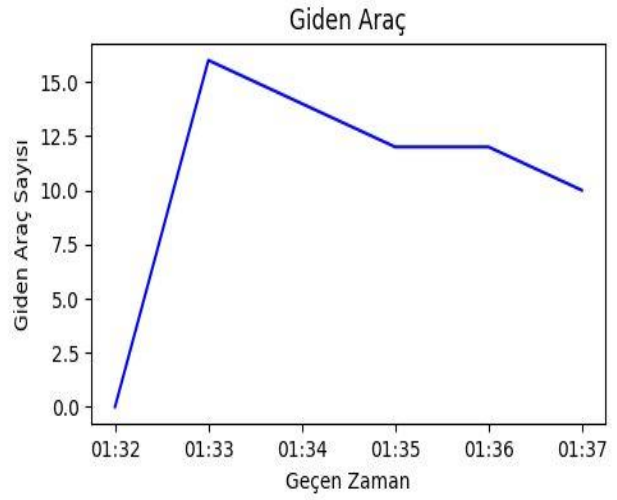
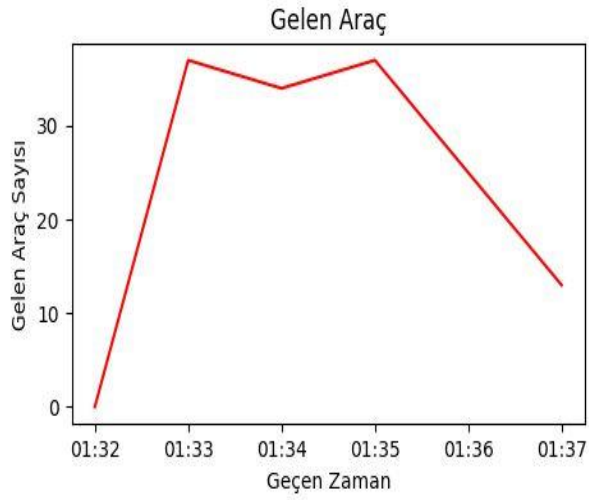
```
520 cv2.putText(Sonuc, str(Sensor2.Mavi_Arac), (Sensor2.Koordinat1.x+225,150), font, 2, (255,0,0))
521 cv2.putText(Sonuc, str(Sensor1.Yesil_Arac), (Sensor1.Koordinat1.x+200,150), font, 2, (0,255,0))
522 cv2.putText(Sonuc, str(Sensor2.Yesil_Arac), (Sensor2.Koordinat1.x+300,150), font, 2, (0,255,0))
523
524
525 if now < video_bölme_bitis_zamanı : # video_bölme_bitis_zamanı 'na aralığımızı ekledik ve zaman ona yaklaştıkça sürekli arttırıyoruz.
526     cv2.imshow("Sonuc", Sonuc)
527     now = datetime.now()
528 elif video_bölme_bitis_zamanı <= now: #HH:MM:SS
529     zaman = video_bölme_bitis_zamanı.strftime("%H:%M") #datetime string türüne dönüştürüldü
530     gecen_zaman.append(zaman) #diziye aktarıldı
531     video_bölme_bitis_zamanı += timedelta(seconds=aralıkT0saniye)
532     gelen_arac_sayisi.append(Sensor2.Algilanan_Arac_Sayisi)
533     giden_arac_sayisi.append(Sensor1.Algilanan_Arac_Sayisi)
534     gelen_uzun_arac_sayisi.append(Sensor2.Algilanan_Uzun_Arac_Sayisi)
535     giden_uzun_arac_sayisi.append(Sensor1.Algilanan_Uzun_Arac_Sayisi)
536     gelen_kirmizi_arac_sayisi.append(Sensor2.Kirmizi_Arac)
537     giden_kirmizi_arac_sayisi.append(Sensor1.Kirmizi_Arac)
538     gelen_beyaz_arac_sayisi.append(Sensor2.Beyaz_Arac)
539     giden_beyaz_arac_sayisi.append(Sensor1.Beyaz_Arac)
540     gelen_mavi_arac_sayisi.append(Sensor2.Mavi_Arac)
541     giden_mavi_arac_sayisi.append(Sensor1.Mavi_Arac)
542     gelen_sari_arac_sayisi.append(Sensor2.Sari_Arac)
543     giden_sari_arac_sayisi.append(Sensor1.Sari_Arac)
544     gelen_yesil_arac_sayisi.append(Sensor2.Yesil_Arac)
545     giden_yesil_arac_sayisi.append(Sensor1.Yesil_Arac)
546     Sensor1.Algilanan_Arac_Sayisi = 0
547     Sensor2.Algilanan_Arac_Sayisi = 0
548     Sensor1.Algilanan_Uzun_Arac_Sayisi = 0
549     Sensor2.Algilanan_Uzun_Arac_Sayisi = 0
550     Sensor2.Kirmizi_Arac = 0
551     Sensor1.Kirmizi_Arac = 0
552     Sensor2.Beyaz_Arac = 0
553     Sensor1.Beyaz_Arac = 0
554     Sensor2.Mavi_Arac = 0
555     Sensor1.Mavi_Arac = 0
556     Sensor2.Sari_Arac = 0
557     Sensor1.Sari_Arac = 0
558     Sensor2.Yesil_Arac = 0
559     Sensor1.Yesil_Arac = 0
560
561
562 k=cv2.waitKey(60) & 0xff #esc ye bastığında program kapanır
563 if k == 27 :
564     for n in gelen_arac_sayisi:
565         if gelen_en_kucuk > n:
566             gelen_en_kucuk = n
567             gelen_index_kucuk = i
568         if gelen_en_buyuk < n:
569             gelen_en_buyuk = n
570             gelen_index_buyuk = i
571     i = i + 1
572     print("Gelen En Büyük Sayı :", gelen_en_buyuk, "\n Gelen En Küçük Sayı :",gelen_en_kucuk)
573     print("En Büyük index :", gelen_index_buyuk, " :", gelen_arac_sayisi[gelen_index_buyuk], "\n En Küçük index :",gelen_index_kucuk, " :",gelen_arac_sayisi[gelen_index_kucuk])
574     print("GELEN Araç yoğunluğunun en fazla olduğu aralık :", gecen_zaman[gelen_index_buyuk-1], " ile ", gecen_zaman[gelen_index_buyuk], " arasında")
575     for m in giden_arac_sayisi:
576         if giden_en_kucuk > m:
577             giden_en_kucuk = m
578             giden_index_kucuk = j
579         if giden_en_buyuk < m:
580             giden_en_buyuk = m
581             giden_index_buyuk = j
582     j = j + 1
583     print("Giden En Büyük Sayı :", giden_en_buyuk, "\n Giden En Küçük Sayı :",giden_en_kucuk)
584     print("En Büyük index :", giden_index_buyuk, " :", giden_arac_sayisi[giden_index_buyuk], "\n En Küçük index :",giden_index_kucuk, " :",giden_arac_sayisi[giden_index_kucuk])
585     print("GİDEN Araç yoğunluğunun en fazla olduğu aralık :", gecen_zaman[giden_index_buyuk-1], " ile ", gecen_zaman[giden_index_buyuk], " arasında")
586
587     print(gecen_zaman)
588     print(gelen_arac_sayisi)
589     print(giden_arac_sayisi)
590     for i in range(0,len(gecen_zaman)): #dizinin uzunluğunu buluyoruz.
591         planSheet.write(i+1,0,gecen_zaman[i]) # (satır sırası, sütun sırası, değer) zaman sütunu 0. sütunda kaldığı için sadece satırları arttırdım
592
```

```

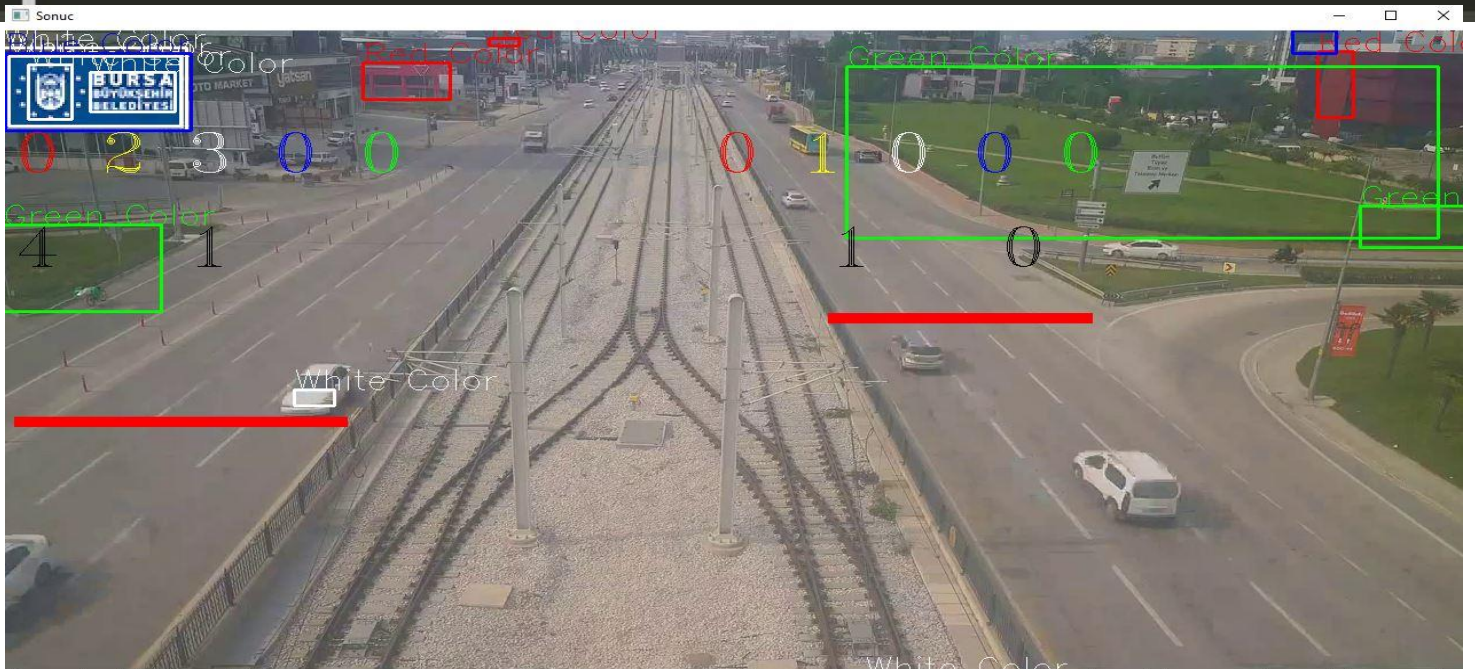
586
587 print(gecen_zaman)
588 print(gelen_arac_sayisi)
589 print(giden_arac_sayisi)
590 for i in range(0,len(gecen_zaman)): #dizinin uzunluğunu buluyoruz.
591     planSheet.write(i+1,0,gecen_zaman[i]) # (sıra sırası, sütun sırası, değer) zaman sütunu 0. sütunda kaldığı için sadece satırları arttırdım
592
593 for i in range(0,len(gelen_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
594     planSheet.write(i+1,1,gelen_arac_sayisi[i]) # Gelen araç sütunu 1. sütunda kaldığı için sadece satırları arttırdım
595
596 for i in range(0,len(gelen_uzun_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
597     planSheet.write(i+1,2,gelen_uzun_arac_sayisi[i]) # Gelen uzun araç sütunu 2. sütunda kaldığı için sadece satırları arttırdım
598
599 for i in range(0,len(giden_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
600     planSheet.write(i+1,3,giden_arac_sayisi[i]) # Giden araç sütunu 3. sütunda kaldığı için sadece satırları arttırdım
601
602 for i in range(0,len(giden_uzun_arac_sayisi)): #dizinin uzunluğunu buluyoruz.
603     planSheet.write(i+1,4,giden_uzun_arac_sayisi[i]) # Giden uzun araç sütunu 4. sütunda kaldığı için sadece satırları arttırdım
604
605 for i in range(0,len(gelen_beyaz_arac_sayisi)):
606     planSheet.write(i+1,5,gelen_beyaz_arac_sayisi[i])
607
608 for i in range(0,len(giden_beyaz_arac_sayisi)):
609     planSheet.write(i+1,6,giden_beyaz_arac_sayisi[i])
610
611 for i in range(0,len(gelen_kirmizi_arac_sayisi)):
612     planSheet.write(i+1,7,gelen_kirmizi_arac_sayisi[i])
613
614 for i in range(0,len(giden_kirmizi_arac_sayisi)):
615     planSheet.write(i+1,8,giden_kirmizi_arac_sayisi[i])
616
617 for i in range(0,len(gelen_mavi_arac_sayisi)):
618     planSheet.write(i+1,9,gelen_mavi_arac_sayisi[i])
619
620 for i in range(0,len(giden_mavi_arac_sayisi)):
621     planSheet.write(i+1,10,giden_mavi_arac_sayisi[i])
622
623 for i in range(0,len(gelen_sari_arac_sayisi)):
624     planSheet.write(i+1,11,gelen_sari_arac_sayisi[i])
625
626 for i in range(0,len(giden_sari_arac_sayisi)):
627     planSheet.write(i+1,12,giden_sari_arac_sayisi[i])
628
629 for i in range(0,len(gelen_yesil_arac_sayisi)):
630     planSheet.write(i+1,13,gelen_yesil_arac_sayisi[i])
631
632 for i in range(0,len(giden_yesil_arac_sayisi)):
633     planSheet.write(i+1,14,giden_yesil_arac_sayisi[i])
634
635
636 planWorkBook.close()
637 plt.figure(figsize=(20,10))
638
639 plt.subplot(2,2,1)
640 plt.plot(gecen_zaman,gelen_arac_sayisi,color="r")
641 plt.xlabel("Geçen Zaman")
642 plt.ylabel("Gelen Araç Sayısı")
643 plt.title("Gelen Araç")
644
645 plt.subplot(2,2,2)
646 plt.plot(gecen_zaman,giden_arac_sayisi,color="blue")
647 plt.xlabel("Geçen Zaman")
648 plt.ylabel("Giden Araç Sayısı")
649 plt.title("Giden Araç")
650 plt.show()
651
652 break
653
654 Video_Okuyucu.release() #dosyayı serbest bırakma, sürekli tutmama kısaca kapatmak gibi
655 cv2.destroyAllWindows() #opencv pencerelerini yok ediyoruz
656

```


Figure 1



```
PS C:\Users\umutb\Desktop\car-counting-with-python-master2> & C:/Users/umutb/AppData/Local/Programs/Python/Python38-32/Scripts/python.exe car-counting-with-python-master2.py
Gelen En Büyük Sayı : 37
Gelen En Küçük Sayı : 0
En Büyük index : 1 : 37
En Küçük index : 0 : 0
GELEN Araç yoğunluğunun en fazla olduğu aralık : 01:32 ile 01:33 arasındadır
Giden En Büyük Sayı : 16
Giden En Küçük Sayı : 0
En Büyük index : 1 : 16
En Küçük index : 0 : 0
GİDEN Araç yoğunluğunun en fazla olduğu aralık : 01:32 ile 01:33 arasındadır
['01:32', '01:33', '01:34', '01:35', '01:36', '01:37']
[0, 37, 34, 37, 25, 13]
[0, 16, 14, 12, 12, 10]
```



4. KAYNAKLAR (IEEE)

- [1]https://www.youtube.com/watch?v=YV9dHMs4z8c&ab_channel=ERAYAKTOKLUK
- [2]<https://github.com/celalaygar/car-counting-with-python>
- [3]https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html
- [4]<https://www.datascienceearth.com/numpyda-olasilik-dagilimleri-normal-dagilim-binom-dagilimi-poisson-dagilimi-bolum-11/>
- [5]https://github.com/markjay4k/Mask-RCNN-series/blob/master/Mask_RCNN%20Install%20Instructions.ipynb
- [6]https://www.youtube.com/watch?v=tcu4pr948n0&ab_channel=AugmentedStartups
- [7]<https://mobesekamerasi.com/bursa-buttim-kavsagi-canli-mobese-izle/>
- [8]<https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b>
- [9]https://www.youtube.com/watch?v=8m8m4oWsp8M&ab_channel=Pysource
- [10]https://www.youtube.com/watch?v=UdCSiZR8xYY&t=181s&ab_channel=PythonSimplified

ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : UMUT BUĞRA TER
Doğum tarihi ve yeri : 27/05/2001 - ANKARA
e-posta : umutbugra.ter.9@hotmail.com

Öğrenim Bilgileri

| Derece | Okul/Program | Yıl |
|--------|--|-----------|
| Lisans | Balıkesir Üniversitesi/Bilgisayar Mühendisliği | 2019-2023 |
| Lise | Çankaya Lisesi | 2015-2019 |