

T.C.
BALIKESİR ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI



GÖRÜNTÜ İŞLEME YÖNTEMİ İLE SUÇLU TESPİTİ
VE
UYARI SİSTEMİ

GİZEM SOLMAZ

LİSANS TEZİ

Jüri Üyeleri : Dr. Öğr. Üyesi Gültekin KUVAT

BALIKESİR, OCAK - 2022

ÖZET

GÖRÜNTÜ İŞLEME İLE SUÇLU TESPİTİ VE UYARI SİSTEMİ BİTİRME PROJESİ GİZEM SOLMAZ

(TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ GÜLTEKİN KUVAT)

BALIKESİR, OCAK - 2022

Kaçak suçluların ilgili ekipler tarafından bulunmaları uzun sürebiliyor. Hedefim suçluların bulundukları şehri ya da ülkeyi terk etme durumlarında ulaşımı sağlayabilmek için tren garları, otobüs terminalleri, havaalanları veya iskeleler, aynı şekilde kaçak suçlunun gidebilme ihtimalinin yüksek olduğu market zincirler, büfeler, petrol ofisleri, otel gibi bir çok yerin kameralarından suçlunun tespitiyle birlikte ilgili birimlere bildiri gönderilmesidir. Kameradan gelen anlık görüntü verisi ile suçluların verilerinin kıyaslanıp suçlu tespiti sağlandığı takdirde sesli olarak uyarı sağlanacaktır.

ABSTRACT

CRIMINAL DETECTION AND WARNING SYSTEM BY IMAGE PROCESSING GRADUATION PROJECT

GİZEM SOLMAZ

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

BALIKESİR, JANUARY - 2022

It may take a long time for illegal criminals to be found by the relevant teams. My goal is to reach the relevant units from the cameras of many places such as train stations, bus terminals, airports or piers, as well as market chains, kiosks, oil offices, hotels, where the illegal criminal is likely to go, in order to provide transportation in case the criminals leave the city or country where they are located. notification is sent. If the snapshot data from the camera and the data of the criminals are compared and the criminal is detected, an audible warning will be provided.

İçindekiler

ÖZET	ii
ABSTRACT	iii
ÖNSÖZ	2
GİRİŞ	3
1. Dosya İçinden Veri Kıyaslaması	3
1.1. Çıktılar	4
2. Kameradan Gelen Görüntü ile Dosyadan Gelen Görüntünün Kıyaslanması	5
2.1 Çıktılar	7
Anlık görüntü ile veri seti içerisinde ki kıyaslama sonucu:	7
Göz bebeği, kaşın yapısı ve şakakların konumundan olan eşleşme sonucu:	8
Veri seti içerisindeki kişi fotoğrafını çıkardığımızda:	8
Uygulamadan çıkış yapıldığı takdirde oluşan .csv dosya yapısı	8
HEDEFLERİM	9
KAYNAKÇA	10

ÖNSÖZ

Projemi yapma amacım kaçak suçluların günümüz şartlarında bile haftalar, aylar gibi uzun zaman zarflarında bulunmasına hatta kimi zaman hiç bulunamama durumlarına son vermektir. Kaçan bir suçlunun bulunduğu şehri hatta ülkeyi terk edebilmek için otobüs, uçak gibi yolculuk biletlerine erişim sağlayabileceği mekanların kameralarıyla suçlu tespiti yapılabilir. Suçlu tespit edildiği takdirde ise ilgili birimlere bildiri gönderilir. Suçlu kaçmak yerine bulunduğu yerde saklanmak istediği takdirde ise en temel ihtiyaçlarını karşılamak için yani yemek, barınma gibi ihtiyaçlarını karşılayabilmek için market zincirlerde, büfe, geceleri alışverişe elverişli olan petrol ofisleri, barınmak için ise lüks oteller yerine daha orta seviyeli dikkat çekmeyecek otel kameralarından da suçlunun tespitini rahatça sağlayıp, tespit edildiği takdirde ise sesli bir uyarı sağlayıp durumu bildirmektir.

GİRİŞ

Suçluların kimlik tespitlerini sağlayabilmek kullanılabilecek birden fazla kütüphane vardır. Ben face_recognition kütüphanesini kullanacağım.

1. Dosya İçinden Veri Kıyaslaması

```
1 import cv2 as cv
2 import numpy as np
3 import face_recognition
4
5 imgElon=face_recognition.load_image_file("images/elonmusk.jpg")
6 imgElon=cv.cvtColor(imgElon,cv.COLOR_BGR2RGB)
7 imgTest=face_recognition.load_image_file("images/elon Test.jpg")
8 imgTest=cv.cvtColor(imgTest,cv.COLOR_BGR2RGB)
9
10 faceLoc=face_recognition.face_locations(imgElon)[0]
11 encodeElon=face_recognition.face_encodings(imgElon)[0]
12 cv.rectangle(imgElon,(faceLoc[3],faceLoc[0]),(faceLoc[1],faceLoc[2]),(255,0,255),2)
13 #print(faceLoc) fotoda ki yüzün lokasyonunu aldı
14 faceLoc=face_recognition.face_locations(imgTest)[0]
15 encodeTest=face_recognition.face_encodings(imgTest)[0]
16 cv.rectangle(imgTest,(faceLoc[3],faceLoc[0]),(faceLoc[1],faceLoc[2]),(255,0,255),2)
17
18 results=face_recognition.compare_faces([encodeElon],encodeTest)
19 faceDis=face_recognition.face_distance([encodeElon],encodeTest)
20 #print(results,faceDis)
21 cv.putText(imgTest,f'{results}{round(faceDis[0],2)}',(20,20),cv.FONT_HERSHEY_COMPLEX_SMALL,1,(200,12,75),1)
22
23 cv.imshow("elon musk",imgElon)
24 cv.imshow("elon test",imgTest)
25 cv.waitKey(0)
```

Şekil 1

Önce lazım olan kütüphaneleri import edildi.

Şekil 1 de 5. ve 7. Satırlarda ki “face_recognition.load_image_file” komutu ile imgElon ve imgTest değişkenlerine veri olarak kullanacağımız resimlerin yollarını belirledik.

Şekil 1 de 6. ve 8. Satırlarda ki cv.cvtColor a giden 2 parametre var. Bunlardan birisi img yani renk uzayını değiştirmek istediğimiz fotoğraf ikinci parametre ise çevirmek istediğimiz renk uzayını belirtilir.

Şekil 1 de 10. ve 14. Satırlarda “face_recognition.face_locations” ile fotoğraflarda ki yüzün konumunu belirler.

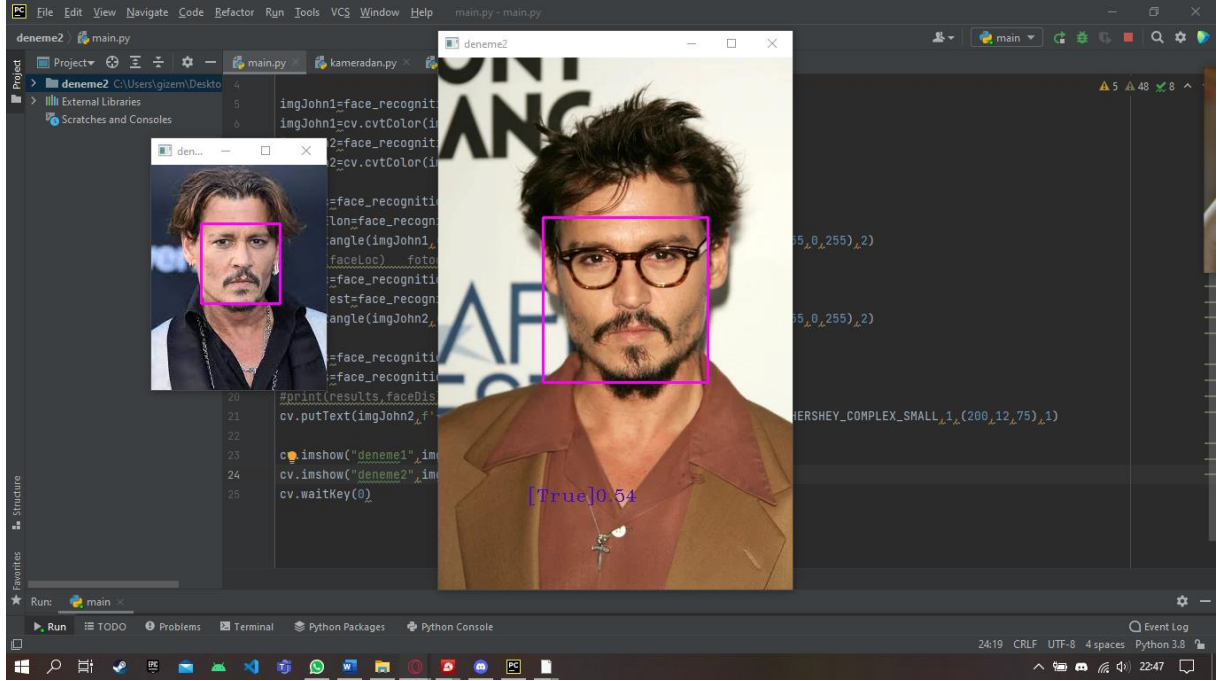
Şekil 1 de 11. ve 15. Satırlarda ki “face_recognition.face_encodings” ile bilgisayarın anlayabileceği dile yani matris şekline çevirmiş olduk.

Şekil 1 de 12. ve 16. Satırlarda bulunan yüz “rectangle” ile dörtgen içerisine alınmış oldu.

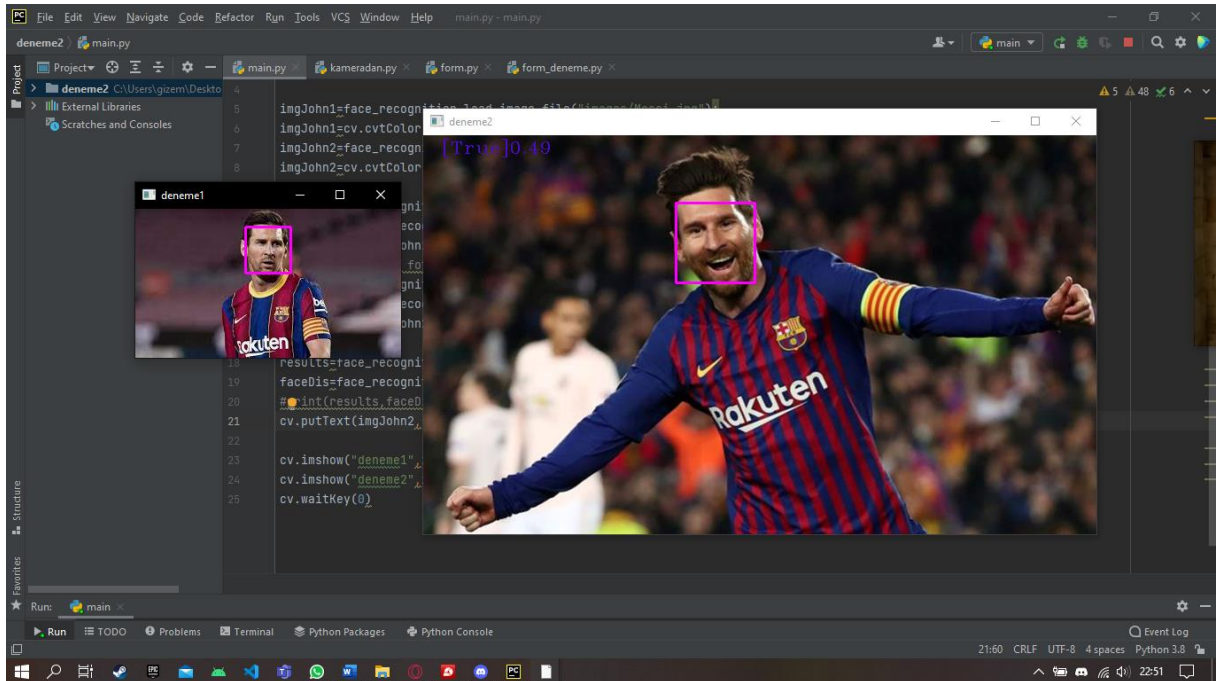
Şekil 1 de 18. satırda “face_recognition.compare_faces” ile 2 parametre kullanılır. Bu parametreler daha önce face_encodings ile verileri matris olarak değişkenlere atamış olduğumuz verilerdir.

Şekil 1 de 19. satırda “face_recognition.face_distance” karşılaştırılan 2 veri arasındaki öklid mesafesine bakılır ve bu mesafeye göre aralarında ki benzerlik oranı ortaya çıkar.

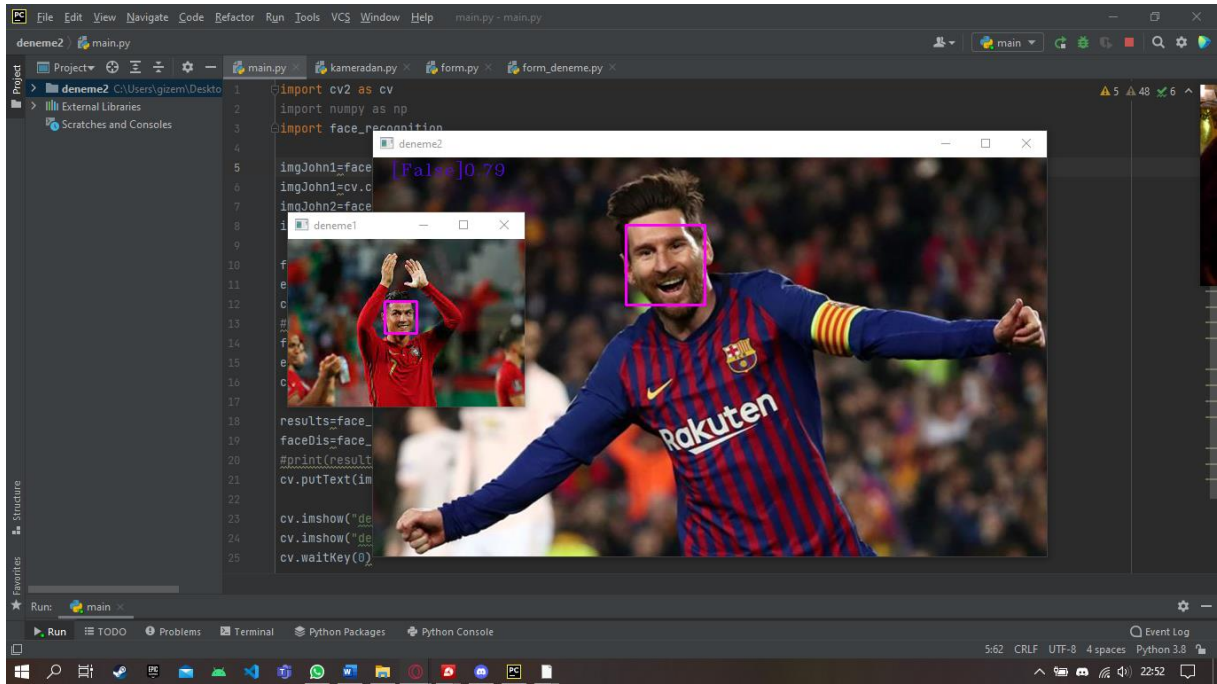
1.1. Çıktılar



Şekil2



Şekil 3



Şekil 4

2. Kameradan Gelen Görüntü ile Dosyadan Gelen Görüntünün Kıyaslanması

```

1 import cv2 as cv
2 import numpy as np
3 import face_recognition
4 import os
5 import csv
6 import datetime
7 import json
8 from urllib.request import urlopen
9
10 dosyaYolu = "kamera_icin"
11 images=[]
12 classNames=[]
13 verilistesi= os.listdir(dosyaYolu)
14 for cl in verilistesi:
15     curImg=cv.imread(f"{dosyaYolu}/{cl}")
16     images.append(curImg)
17     classNames.append(os.path.splitext(cl)[0])
18 def findEncodings(dosyaIci):
19     yeniBosYuzListesi=[]
20     for dosyaIci_yuzBulma in dosyaIci:
21         dosyaIci_yuzBulma = cv.cvtColor(dosyaIci_yuzBulma, cv.COLOR_BGR2RGB)
22         dosyaIci_bulunanYuz = face_recognition.face_encodings(dosyaIci_yuzBulma)[0]
23         yeniBosYuzListesi.append(dosyaIci_bulunanYuz)
24     return yeniBosYuzListesi
25 encodeListKnow=findEncodings(images)

```

Şekil 5

Şekil 5 de ilk 8 satırda kullanacağım kütüphaneleri import ettik.

Şekil 5 de 10-13 satırları arasında ulaşmak istediğimiz veri setinin yolu belirlenir. Ve bu yolu “dosyaYolu” değişkenine atadık. Os.listdir komutu ile dosya içindeki verileri istediğimiz doğrultusunda terminale .jpg/.png şeklinde yazdırabiliyoruz.

Şekil 5 de 14-17 satırları arasındaki açılan for döngüsü ile veri setimiz içinde ki verileri tek tek gezip uzantılarından kurtularak yine isteğimiz doğrultusunda “print(classNames)” ile terminale temiz bir şekilde yazdırabiliyoruz.

Şekil 5 de 18-24 satırları arasında oluşturmuş olduğum boş dizi içerisinde veri setinin içinde ki yüz verilerini tutmuş oldum.

```
28 while True:
29     success, kamera = cap.read()
30     boyutlandırilmisGoruntu=cv.resize(kamera,(0,0),None,0.25,0.25)
31     boyutlandırilmisGoruntu = cv.cvtColor(boyutlandırilmisGoruntu, cv.COLOR_BGR2RGB)
32     faceCurFrame = face_recognition.face_locations(boyutlandırilmisGoruntu)
33     encodesCutFrame = face_recognition.face_encodings(boyutlandırilmisGoruntu,faceCurFrame)
34     for encodeFace,faceLoc in zip(encodesCutFrame,faceCurFrame):
35         matches=face_recognition.compare_faces(encodeListKnow,encodeFace)
36         faceDis=face_recognition.face_distance(encodeListKnow,encodeFace)
37         #print(faceDis)
38         matchIndex=np.argmin(faceDis)
39         if matches[matchIndex]:
40             name=classNames[matchIndex].upper()
41             #print(name)
42             y1, x2, y2, x1=faceLoc
43             y1,x2,y2,x1=y1*4,x2*4,y2*4,x1*4
44             cv.rectangle(kamera,(x1,y1),(x2,y2),(0,0,255),3)
45             cv.rectangle(kamera,(x1,y2-35),(x2,y2),(0,0,0),cv.FILLED)
46             cv.putText(kamera,name,(x1+6,y2-6),cv.FONT_HERSHEY_COMPLEX,1,(0,0,255),2)
47         else:
48             y1, x2, y2, x1 = faceLoc
49             y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
50             cv.rectangle(kamera, (x1, y1), (x2, y2), (0, 0, 0),3)
51             cv.rectangle(kamera, (x1, y2 - 35), (x2, y2), (0,0,0), cv.FILLED)
52             cv.putText(kamera, "Bilinmeyen", (x1 + 6, y2 - 6), cv.FONT_HERSHEY_COMPLEX, 1, (0,0,255), 2)
```

Şekil 6

Şekil 6 de kamera işlemlerine geçiyorum. Burada kameradan gelen anlık görüntüde ilk olarak face_locations ile yüzün konumunu buluyorum daha sonrasında face_encodings ile matris şeklinde bir değişkene atıyorum.

Bu işlemlerden sonra ise kameradan gelen görüntü veri seti içerisinde var mı/yok mu kontrolü için dosya içinde ki verileri tuttuğum “encodeListKnow” değişkeni ve “encodeFace” yani dosya içindeki veriler ile kameradan gelen anlık görüntü arasında kıyaslama yaptırıp if ile de eğer eşleme konusunda başarı sağlıyor ise anlık görüntüde ki kişi yüzünü çerçeve içine alıp veri seti içerisinde ki ad-soyadını çerçeve altına yazdırdık.

Eğer veri seti içinde ki kameradan gelen görüntünün kıyas sonucu olumsuz ise else bloğuna girip kameradan gelen görüntü de ki yüzü çerçeve içine alarak bu sefer çerçeve altına “Bilinmeyen” yazmasını sağladım.

```

54     cv.imshow("Webcam",kamera)
55     if cv.waitKey(1) & 0xFF == ord('q'): # q ile çıkış yapabilirsiniz
56
57         with open(name+".csv", "a", newline="") as f:
58             url = 'http://ipinfo.io/json'
59             response = urlopen(url)
60             konum = json.load(response)
61             yazıcı = csv.writer(f)
62             yazıcı.writerow([name, datetime.datetime.now(),konum])
63         break
64     cv.waitKey(1)
65

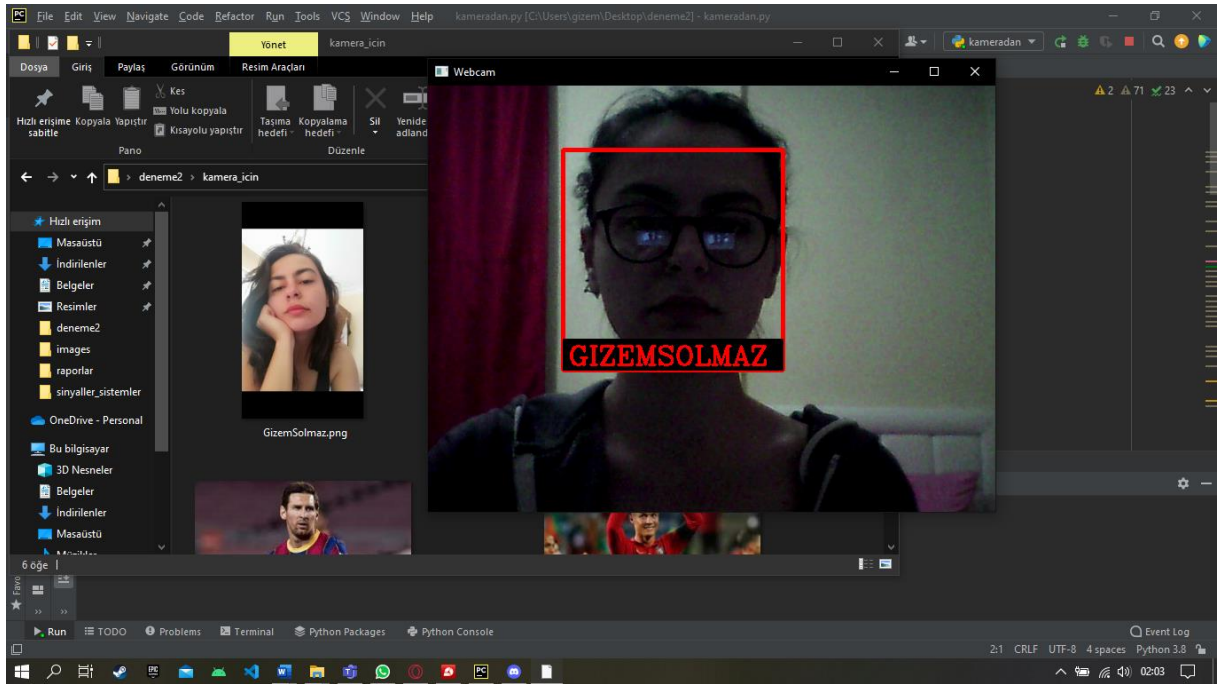
```

Şekil 7

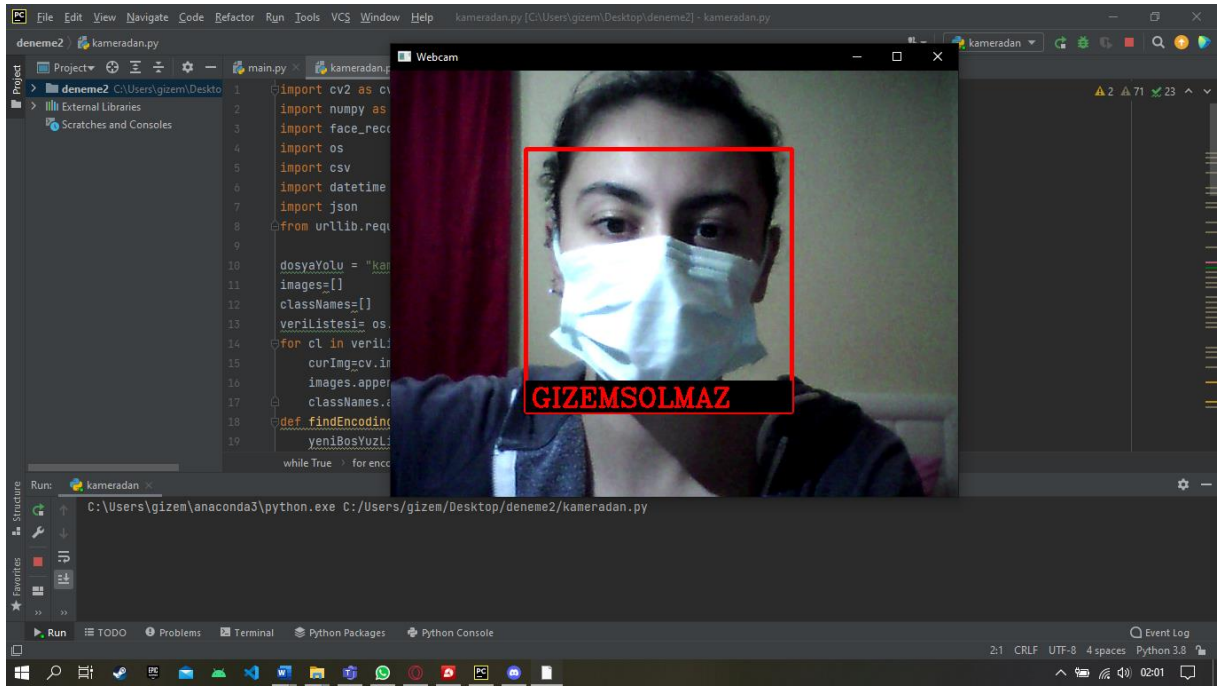
Şekil 7 da 57-63 satırları içerisinde eşlesen verileri elimde tutabilmek için .csv dosyası oluşturmasını sağladım ve bu dosya adını da tespit edilen kişi adı olarak belirledik. Açmış olduğumuz bu .csv içerisinde tespiti sağlanan kişinin adı soyadı dışında tespit edildiği zamanı (gg/aa/yy) şeklinde ve aynı zamanda tespitin sağlandığı mekanın ip adresi, il, posta kodu vs gibi bilgiler ile .csv dosyası içerisine aktarıyorum.

2.1 Çıktılar

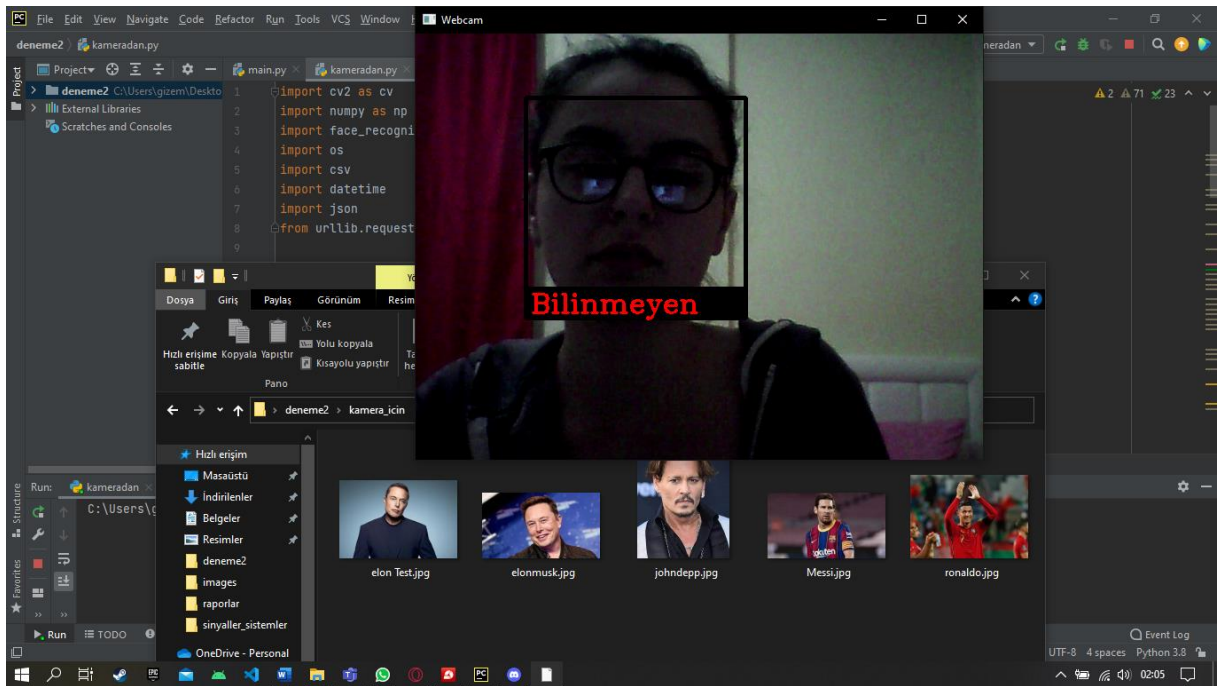
Anlık görüntü ile veri seti içerisinde ki kıyaslama sonucu:



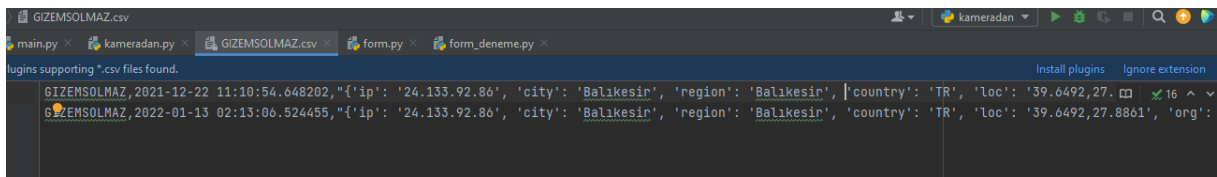
Göz bebeği, kaşın yapısı ve şakakların konumundan olan eşleşme sonucu:



Veri seti içerisindeki kişi fotoğrafını çıkardığımızda:



Uygulamadan çıkış yapıldığı takdirde oluşan .csv dosya yapısı



HEDEFLERİM

- Kameradan gelen anlık görüntü içerisinde birden fazla eşlesen yüz verisi olduğu takdirde eşleşen veri sayısı kadar .csv dosyası oluşturup isimleri ile birlikte oluşturmak/ekleme yapmak istiyorum.
- .csv dosyası içerine girdirmiş olduğum bilgiler üzerinde değişiklikler yapmak istiyorum. Bu değişiklikler eşleşme esnasında ki ip adresi ve konum gibi bazı bilgileri daha düzenli ve daha kullanışlı bir hale getirmek istiyorum.
- .csv dosyasının oluşumunu tetikleyen durum “q” ya basmaktır. Yani uygulamayı kapattığım zaman .csv dosyası oluşmaktadır. Normal şartlarda market, petrol ofisi, otel vs. gibi yerlerde eşleşmenin sağlandığı an dosya oluşturup verileri saklamak istiyorum. Aksi takdirde devamlı olarak eşleşme olup olmadığı manuel olarak kontrol edilmeli ve eşleşme esnasında uygulamanın kapatılması gerekir. Bu durum efektif olmadığı için daha farklı bir yol uygulayacağım.
- Eşleşmenin kesinlik düzeyine göre “Düşük Benzerlik” ve “Yüksek Benzerlik” adlı 2 farklı klasör içerisinde .csv oluşturup, “Yüksek Benzerlik” içerisine veri kaydı olduğunda bilgisayardan uyarı sesinin çıkması sağlanacaktır.

Sizin de yönlendirmeniz ile projemi benim hedeflerim haricinde daha iyi bir şekilde geliştireceğimi düşünüyorum.

.

KAYNAKÇA

<https://pypi.org/project/face-recognition/>

https://google.github.io/mediapipe/solutions/face_detection.html