

**BALIKESİR ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**



**Görüntü İşleme Yöntemi İle Suçlu Tespiti ve  
Uyarı Sistemi**

**BİTİRME PROJESİ**  
**RAPOR-3**

**Dr. Öğr. Üyesi: Gültekin KUVAT**

**Ad Soyad: Gizem SOLMAZ**

**Numara: 201613709028**

**Bilgisayar Mühendisliği Tasarımı**

**04.12.2021**

## GEÇEN HAFTA YAPTIKLARIM - BU HAFTA YAPTIKLARIM

Bir önceki aşama da dosya içindeki fotoğraflarda yüzün konumunu buldum ve bu verileri boş bir dizi içerisinde tuttum. Şimdi yapacağım şey ise bu dizi içerisinde ki veriler ile kameradan gelen görüntü ile karşılaştırma yapacağım.

## BİR ÖNCE Kİ AŞAMADA YAZMIŞ OLDUĞUM KOD BLOĞU

```
main.py x kameradan.py x
1 import cv2 as cv
2 import numpy as np
3 import face_recognition
4 import os
5
6 dosyaYolu = "kamera_icin"
7 images=[]
8 classNames=[]
9 verilistesi= os.listdir(dosyaYolu)
10 #print(verilistesi)
11
12 for cl in verilistesi:
13     curImg=cv.imread(f"{dosyaYolu}/{cl}")
14     images.append(curImg)
15     classNames.append(os.path.splitext(cl)[0])
16 print(classNames)
17
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
```

Artık kamerayı açıp işlemlerime devam edeceğim.

```
29 cap=cv.VideoCapture(0)
30 while True:
31     success, kamera = cap.read()
32     boyutlandirilmisGoruntu=cv.resize(kamera,(0,0),None,0.25,0.25)
33     boyutlandirilmisGoruntu = cv.cvtColor(boyutlandirilmisGoruntu, cv.COLOR_BGR2RGB)
34     faceCurFrame = face_recognition.face_locations(boyutlandirilmisGoruntu)
35     encodesCutFrame = face_recognition.face_encodings(boyutlandirilmisGoruntu,faceCurFrame)
```

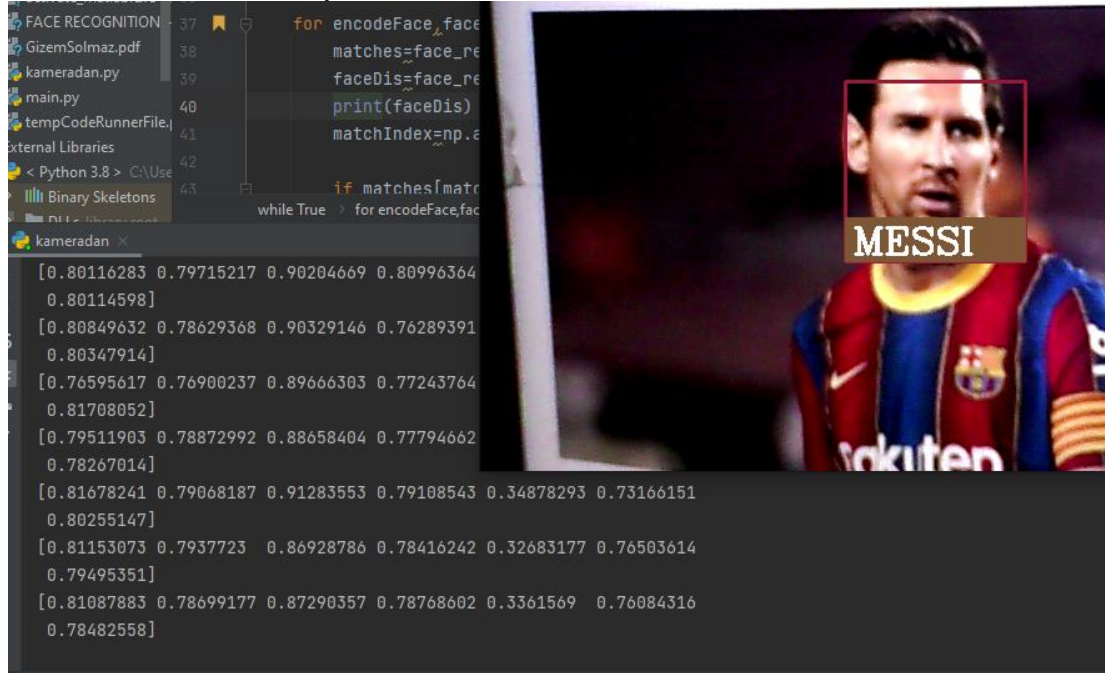
While ile sonsuz bir döngü açıyorum ve artık kalan tüm işlemlerimi bu döngü içinde gerçekleştireceğim. Öncelikle kameramı açıyorum. Daha sonrasında resize ile kameradan gelen görüntüyü yeniden boyutlandırıyorum. Yeniden boyutlandırmış olduğum gelen görüntüyü BGR den RGB ye çeviriyorum. Şimdi ise sadece yüzü bir çerçeve içine alıp “faceCurFrame” içerisinde tutuyorum. Daha sonra çerçeve içinde ki yüzde ki kimlik tespiti için önemli olan hatları bulup “encodesCutFrame” için de tutmuş oldum.

```

37 for encodeFace, faceLoc in zip(encodesCutFrame, faceCurFrame):
38     matches=face_recognition.compare_faces(encodeListKnow, encodeFace)
39     faceDis=face_recognition.face_distance(encodeListKnow, encodeFace)
40     #print(faceDis)
41     matchIndex=np.argmin(faceDis)

```

Eşleşip eşleşmediklerini görebilmek için 38. satırda bilinen yüz kodlarının listesi(encodeListKnow) ve karşılaştırmak için de encodeFace'i kıyasladım. 39. Satırda *face\_distance* kıyaslama sonucu yüzlerin ne kadar benzer olduğunu bulup 40. Satırda bunu terminalde yazdırdım.



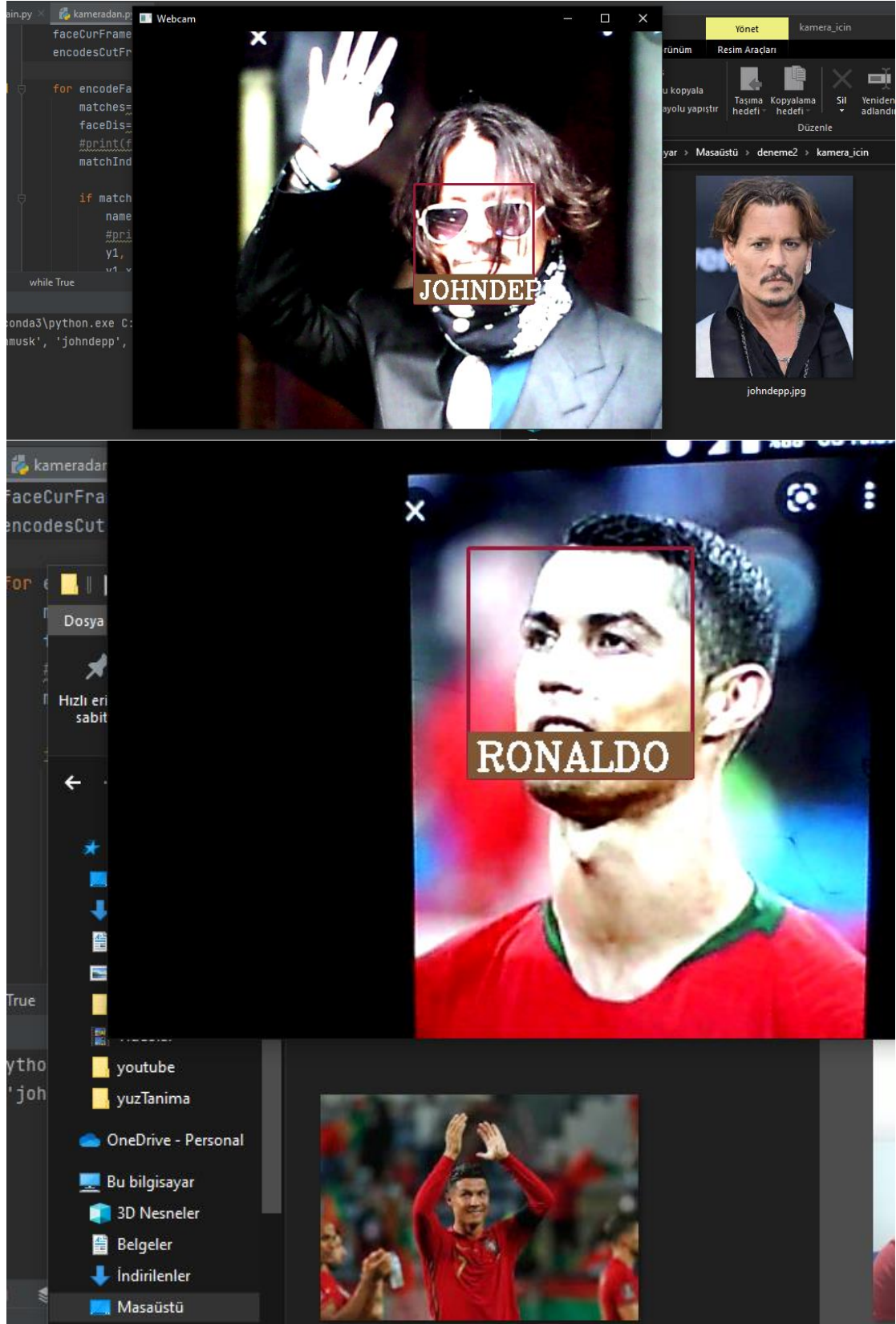
Bulmuş olduğum yüzü çerçeveleyip daha sonra kimlik tespitinden elde edilen ismi yazdırmak için ise aşağıda ki kod bloğunu yazdım.

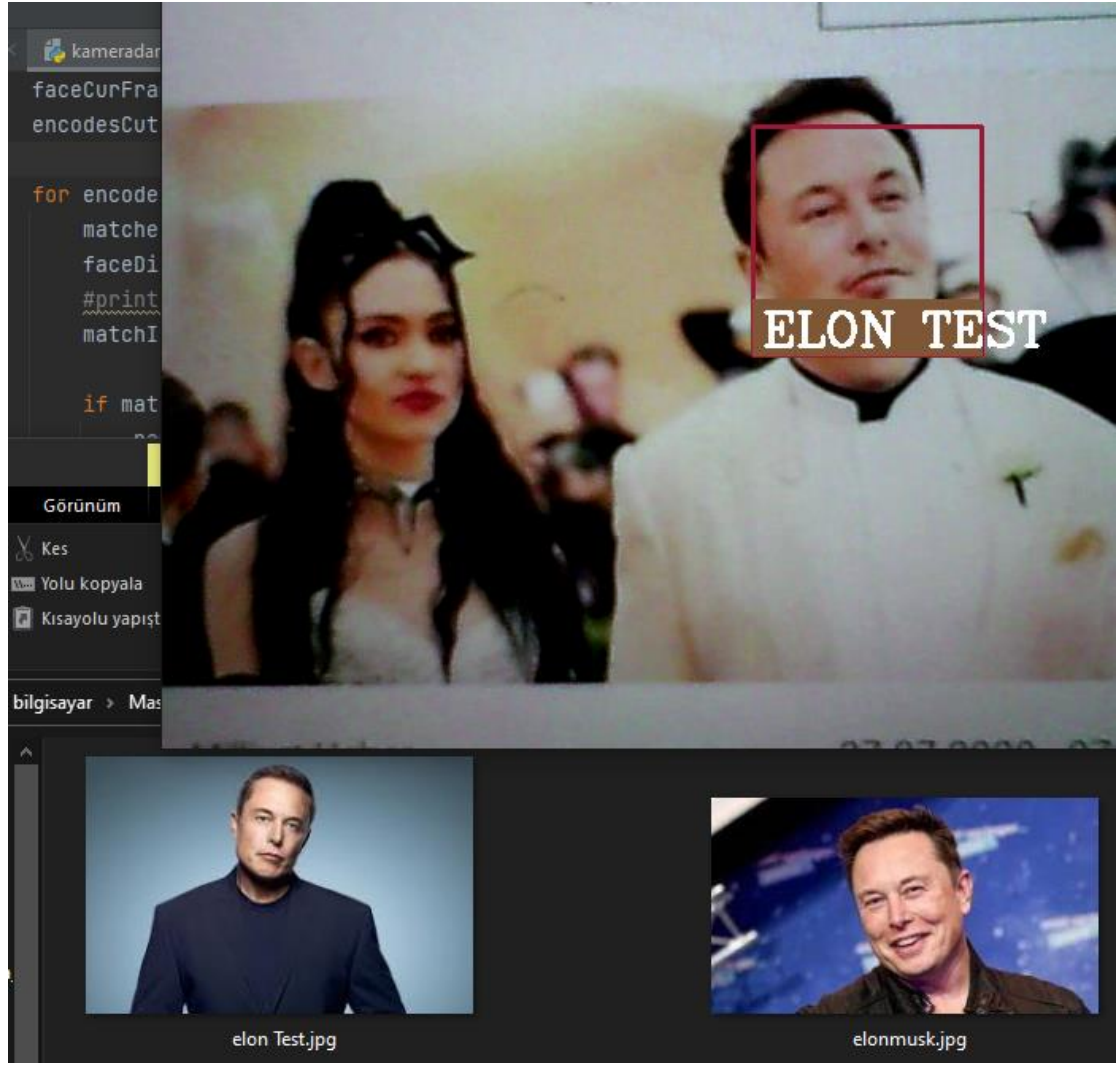
```

43 if matches[matchIndex]:
44     name=classNames[matchIndex].upper()
45     #print(name)
46     y1, x2, y2, x1=faceLoc
47     y1, x2, y2, x1=y1*4, x2*4, y2*4, x1*4
48     cv.rectangle(kamera, (x1, y1), (x2, y2), (54, 26, 153), 2)
49     cv.rectangle(kamera, (x1, y2-35), (x2, y2), (54, 87, 125), cv.FILLED)
50     cv.putText(kamera, name, (x1+6, y2-6), cv.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)

```

Doyanın içinde var olan diğer verilerle olan sonuçlar:





## KAYNAKÇA

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

[https://google.github.io/mediapipe/solutions/face\\_detection.html](https://google.github.io/mediapipe/solutions/face_detection.html)