



Görüntü İşleme Dersi Yüz Tanıma Projesi Raporu

Öğrenciler:

Atakan AYMANKUY 2190656039

Yiğit CAN 2190656026

Danışman:

Doç. Dr. Ahmet SAYGILI

İÇİNDEKİLER

Görüntü İşleme Dersi Yüz Tanıma Projesi Raporu	1
1.GİRİŞ.....	3
2.PROJE AMAÇLARI.....	4
3.KULLANILAN PROGRAMLAR	5
4.PROJE KODU	6
4.PROJE KODU	7
5. PROJE ÇIKTILARI.....	8
6.PROJE DEĞERLENDİRMESİ	9
7.SONUÇ.....	10

1.GİRİŞ

Bu rapor, görüntü işleme dersi kapsamında gerçekleştirilen yüz tanıma projesinin detaylarını içermektedir. Bu proje, yüz tanıma algoritmalarını kullanarak görüntülerdeki yüzleri tespit etmek ve tanımak için bir sistem geliştirmeyi amaçlamaktadır. Rapor, proje amaçları, kullanılan yöntemler, elde edilen sonuçlar ve projenin genel değerlendirmesini içerecektir.

Anahtar Kelimeler: Python, OpenCV, Veri seti, Yapay zeka

2.PROJE AMAÇLARI

Yüz tanıma projesinin temel amaçları aşağıda belirtilmiştir:

- a) Görüntülerdeki yüzleri doğru bir şekilde tespit etmek.
- b) Yüzlerin özelliklerini çıkarmak ve temsil etmek için uygun algoritmaları kullanmak.
- c) Tanınan yüzleri doğru bir şekilde sınıflandırmak ve kimliklerini belirlemek.
- d) Performansı değerlendirmek ve iyileştirme olanaklarını araştırmak.

3.KULLANILAN PROGRAMLAR

Python, kullanımı kolay ve güçlü bir programlama dilidir. Genel amaçlı bir dil olmasının yanı sıra, bilimsel hesaplama, veri analizi ve görüntü işleme gibi alanlarda da popülerdir. OpenCV (Open Source Computer Vision Library), bilgisayarla görü analizi ve makine görü alanlarında kullanılan bir kütüphanedir.

Python ve OpenCV, görüntü işleme uygulamaları geliştirmek için bir araya getirilen güçlü bir kombinasyondur. OpenCV, görüntülerin ve videoların işlenmesi, görüntü özelliklerinin çıkarılması, nesne tanıma, yüz algılama, görüntü segmentasyonu, stereo görüntüleme ve çok daha fazlası gibi bir dizi görüntü işleme işlevini destekler.

OpenCV, Python için bir dizi işlev ve sınıf içeren bir modüldür. Görüntülerin yüklenmesi, kaydedilmesi, dönüştürülmesi, filtrelenmesi, morfolojik işlemlerin uygulanması, kenar tespiti, histogram eşitleme, görüntü piramitleri, kamera kalibrasyonu gibi birçok işlemi gerçekleştirilir.

Python ve OpenCV kombinasyonu, görüntü işleme, robotik, otomasyon, yapay zeka, nesne algılama ve benzeri alanlarda birçok uygulamanın geliştirilmesinde yaygın olarak kullanılmaktadır. Ayrıca, Python ve OpenCV'nin geniş topluluk desteği sayesinde belgeleme, örnek kodlar ve sorun giderme konusunda da kolaylık sağlanmaktadır.

4.PROJE KODU

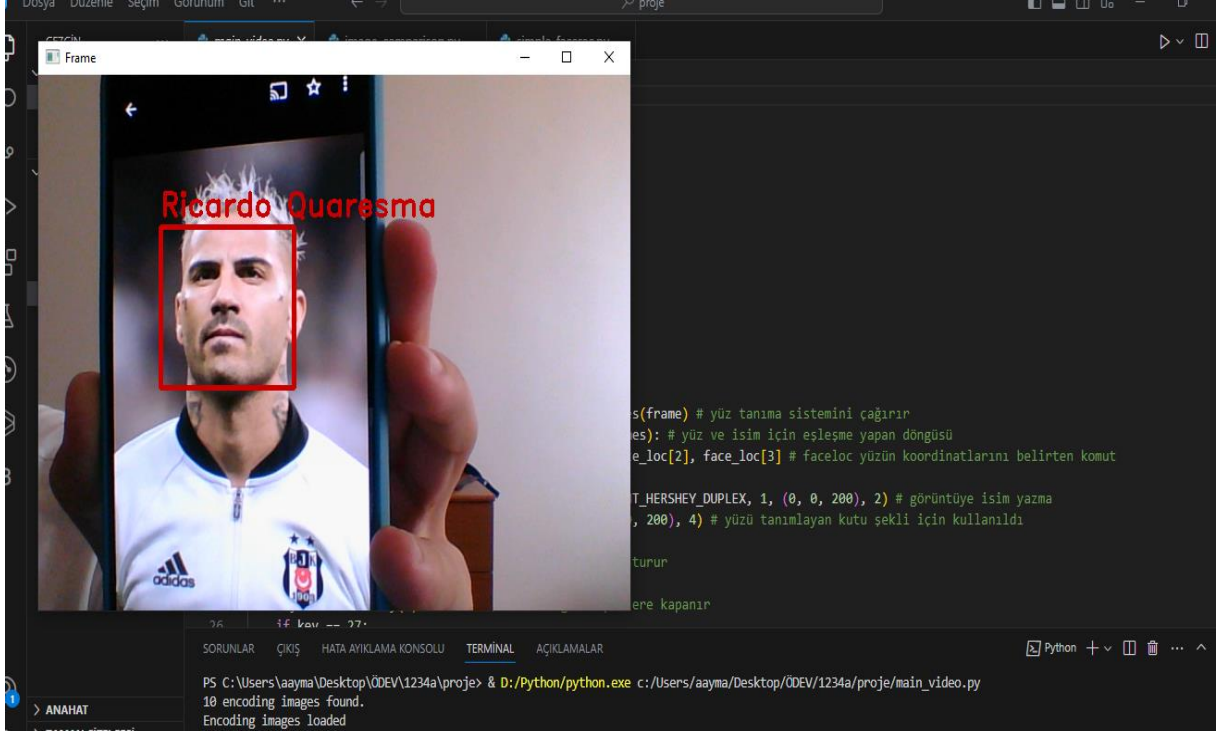
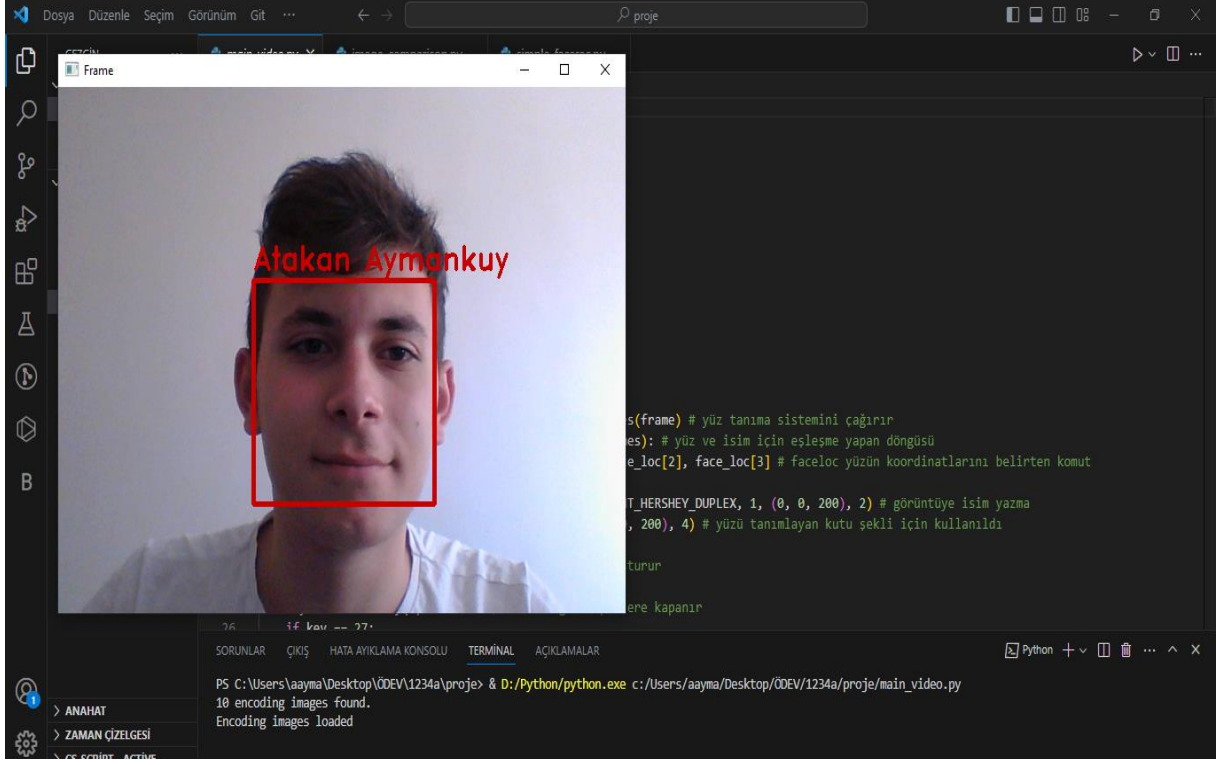
```
1 import cv2
2 import face_recognition
3
4 img = cv2.imread("Messi1.webp")
5 rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
6 img_encoding = face_recognition.face_encodings(rgb_img)[0] # yüzleri algılar ve tanımlar
7
8 img2 = cv2.imread("images/Messi.webp")
9 rgb_img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
10 img_encoding2 = face_recognition.face_encodings(rgb_img2)[0]
11
12 result = face_recognition.compare_faces([img_encoding], img_encoding2) # önceki yüzleri karşılaştırmada kullanılır
13 print("Result: ", result)
14
15 cv2.imshow("Img", img)
16 cv2.imshow("Img 2", img2)
17 cv2.waitKey(0)
18 # bu kod iki farklı görüntüdeki yüzleri algılamak için yazılmıştır
```

```
1 import cv2
2 from simple_facerec import SimpleFacerec
3
4 # klasörden resim çekme
5 sfr = SimpleFacerec()
6 sfr.load_encoding_images("images/")
7
8 # kamera açma
9 cap = cv2.VideoCapture(0)
10
11
12 while True:
13     ret, frame = cap.read()
14
15     # yüz tanıma işlemi yapılıyor
16     face_locations, face_names = sfr.detect_known_faces(frame) # yüz tanıma sistemini çağırır
17     for face_loc, name in zip(face_locations, face_names): # yüz ve isim için eşleşme yapan döngüsü
18         y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2], face_loc[3] # faceloc yüzün koordinatlarını belirten komut
19
20         cv2.putText(frame, name, (x1, y1 - 10), cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 200), 2) # görüntüye isim yazma
21         cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 200), 4) # yüzü tanımlayan kutu şekli için kullanıldı
22
23     cv2.imshow("Frame", frame) # görüntüyü ekranda oluşturur
24
25     key = cv2.waitKey(1) # bir tuşa basıldığında pencere kapanır
26     if key == 27:
27         break
28
29 cap.release()
30 cv2.destroyAllWindows() # bütün pencerelerin kapanma komutu
```

4.PROJE KODU

```
1 import face_recognition
2 import cv2
3 import os
4 import glob
5 import numpy as np # gerekli kütüphaneleri import ettik
6
7 class SimpleFacerec:
8     def __init__(self):
9         self.known_face_encodings = [] # bilinen yüzlerin kodlamasını içerir
10        self.known_face_names = [] #bilinen yüzlerin isimlerini içerir
11
12        # görüntüyü boyutlandırma işlemi
13        self.frame_resizing = 0.25
14
15    def load_encoding_images(self, images_path): #bu dizinde görüntüler, isimler çekilir ve ilgili listeye eklenir
16        """
17        Load encoding images from path
18        :param images_path:
19        :return:
20        """
21        # görüntü yükleme
22        images_path = glob.glob(os.path.join(images_path, "*.jpg"))
23
24        print("{} encoding images found.".format(len(images_path)))
25
26        # görüntü ve isimlerin saklanması işlemi
27        for img_path in images_path:
28            img = cv2.imread(img_path)
29            rgb_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
30
31            # başlangıç aşamasından sadece dosya adı çekilir
32            basename = os.path.basename(img_path)
33            (filename, ext) = os.path.splitext(basename)
34
35            img_encoding = face_recognition.face_encodings(rgb_img)[0]
36
37            # 1.satırdaki kod 2.satırdakine görüntü kodlamasını ekler
38            self.known_face_encodings.append(img_encoding)
39            self.known_face_names.append(filename)
40            print("Encoding images loaded")
41
42    def detect_known_faces(self, frame):
43        small_frame = cv2.resize(frame, (0, 0), fx=self.frame_resizing, fy=self.frame_resizing)
44        # mevcut video karesindeki tüm yüzleri ve kodlamalarını bulmak için yazılır
45        # renk formatına dönüştürme işlemi
46        rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
47        face_locations = face_recognition.face_locations(rgb_small_frame)
48        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
49
50        face_names = []
51        for face_encoding in face_encodings:
52            # yüzlerin eşleşip eşleşmediğini kontrol et
53            matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
54            name = "Unknown" #eşleşmediği durumda ekrana unknown yaz
55
56        # bu kod parçası, bir yüz kodlamasının bilinen yüz kodlamaları ile karşılaştırılarak
57        # en iyi eşleşmenin bulunmasını ve ilgili yüz için bir isim belirlenmesini sağlar.
58        face_distances = face_recognition.face_distance(self.known_face_encodings, face_encoding)
59        # yüz kodlamaları arası mesafeyi hesaplar ve kaydeder
60        best_match_index = np.argmin(face_distances) #en iy eşleşen yüzün konumunu belirtir
61        if matches[best_match_index]: #değerin True olup olmadığı kontrol edilir
62            name = self.known_face_names[best_match_index] #eşleşen en iyi yüzün ismini ifade eder
63            face_names.append(name) # elde edilen değeri listeye ekler. bu işlem eklenen her yüz için devam eder sonuçları saklar
64
65        # hızlı bir şekilde çerçeveyi yeniden boyutlandırmak için koordinatları düzenlemek için numpy dizisine dönüştürür
66        face_locations = np.array(face_locations)
67        face_locations = face_locations / self.frame_resizing
68        return face_locations.astype(int), face_names
69
```

5. PROJE ÇIKTILARI



6.PROJE DEĞERLENDİRMESİ

Proje, görüntü işleme dersi için önemli bir uygulama alanını kapsamaktadır. Yüz tanıma, güvenlik sistemleri, sosyal medya etiketleme ve kişiye özel deneyimler gibi birçok alanda kullanılan bir teknolojidir. Proje, yüz tanıma algoritmalarının temel prensiplerini anlamak, uygulamak ve iyileştirmek için iyi bir fırsat sağlamıştır. Geliştirilen sistem, gerçek dünya senaryolarında kullanılabilirliğini artırmak için daha fazla çalışma gerektirmektedir.

7.SONUÇ

Bu rapor, görüntü işleme dersi kapsamında gerçekleştirilen yüz tanıma projesinin detaylarını sunmuştur. Proje, yüz tanıma algoritmalarını kullanarak görüntülerdeki yüzleri tespit etme ve tanıma yeteneklerini göstermektedir. Projenin sonuçları, yüz tespiti ve tanıma konularında başarılı olduğunu göstermektedir. Geliştirilen sistem, daha geniş veri setleri ve performans iyileştirmeleri ile daha da geliştirilebilir. Yüz tanıma teknolojisi, gelecekteki uygulamalarda daha da yaygınlaşacak ve kullanıcı deneyimini artıracaktır.