###### 

###### 

###### MERSİN ÜNİVERSİTESİ

**Mühendislik Fakültesi**

**Bilgisayar Mühendisliği Bölümü**

**2021-2022 Güz Yarıyılı**

**155-7001 - BİTİRME ÖDEVİ**

## Final Proje Raporu

**İlayda ÜNAL**

**(16155051)**

**Ödev Konusu:** Yüz Tanıma ve Duygu Analizi

**i**

**ÖZET**

Günümüzde hızla gelişen teknolojiyle birlikte; yapay zeka, makine öğrenimi ve biyometrik teknolojilerde bir yükseliş gözlemleniyor. Bununla beraber artan erişilebilirlik ve doğruluk sunan yüz tanıma teknolojileri de yaygınlaşarak hayatımızın her alanında yer almaya başladı.

En basit tanımıyla yüz tanıma sistemlerini; kişilerin yüz yapılarını bir şifreleme sistemi olarak değerlendiren ve bunu güvenlik amacıyla kullanan sistemler olarak açıklayabiliriz.

Duygu analizi ise, yüz ifadesi gibi psikolojik durumların sistem tarafından değerlendirilmesi olarak açıklanabilir. Bir yüz ifadesinde aynı anda birkaç ifade görülebilir. Bu nedenle en ufak mimik ve ayrıntı bile önemlidir. Bu sistemlerde sınıflama ve özellik çıkarma işlemlerinin son derece önemli olduğunu bu noktada anlayabiliriz.

Yapılan bu çalışmada, HOG, CNN gibi yapay sinir ağları kullanılarak görüntülerden insanların yüz tespiti yapılıp, bu yüzlerdeki ifadeler üzerinden duygu analizleri yapılacaktır.

**ABSTRACT**

With the rapidly developing technology today; Artificial intelligence, machine learning and biometric technologies are on the rise. In addition, facial recognition technologies that offer increased accessibility and accuracy have become widespread and have begun to take place in all areas of our lives.

In the simplest definition, face recognition systems; We can describe it as systems that evaluate people's facial structures as an encryption system and use it for security purposes.

Sentiment analysis is; It can be explained as the evaluation of psychological states such as facial expression by the system. Several expressions can be seen simultaneously in a facial expression. For this reason, even the smallest gestures and details are important. At this point, we can understand that classification and feature extraction processes are extremely important in these systems.

In this study, people's faces will be detected from images using artificial neural networks such as HOG and CNN, and emotion analysis will be made on the expressions on these faces.

**ii**

**İÇİNDEKİLER**

KAPAK……………………………………………………………………………………………………...i

ÖZET………………………………………………………………………………………………………...ii

BÖLÜM 1. GİRİŞ…………………………………………………………………………………………...1

BÖLÜM 2. KULLANILAN TEKNOLOJİLER……………………………………………………………2

2.1.Materyal-Metot……….…………………………………………………………...…………………2

2.2.Sistemin Genel Yapısı………………………………………………………………………………..3

BÖLÜM 3.YÜZ TANIMA UYGULAMASI......…………………………………………………………..4

3.1. Veri Seti Oluşturma…………………………………………………………………………......4

3.2. Eğitim…………………………………………………………………………………………...8

3.3. Yüz Tanıma ve Duygu Analizi...……………………………………………………………....11

BÖLÜM 4. SONUÇ………………………………………………………………………………………..18

**iii**

**BÖLÜM 1. GİRİŞ**

Günümüzde yüz tanıma teknolojileri, telefonumuzun kilidini açmamızdan, havaalanında güvenlikten geçmenize kadar hayatımızın her alanına girmiş bulunmaktadır. Yüz tanıma; kişinin yüz ayrıntıları yakalayan, analiz eden ve karşılaştıran bir teknolojidir. Bu teknolojide kullanılan yüz algılama işlemi ise sistemlerin bir dizi görüntü ve videodaki insan yüzlerini tespit etmesini ve bulmasını sağlayan önemli bir adımdır. Fotoğraf ve video verilerinin çokluğu yüz tanıma teknolojilerinin çalışabilmesini sağlamak için her türlü dataset’i bize sunuyor. Teknolojiler; sosyal medya, akıllı telefonlarımız, yüksek kapasiteli kameralar ve buna benzer diğer uygulamalar tarafından ortaya çıkan milyarlarca görsel veriyi analiz ediyor. Bir yüzün içerdiği analog bilgiyi, kişinin benzersiz yüz özelliklerine göre bir dizi dijital bilgiye dönüştürüyor. Bu verileri kullanarak yüz eşleştirme işlemi; makine öğrenmesi ve yapay zeka yetenekleri ile iki yüzün aynı kişiye ait olup olmadığını doğruluyor.

Duygu analizi; insan yüzünü tarayarak yüz ifadelerinden duyguları tanıma işlemi olarak açıklanabilir. Duygular tanımlarken 2 tür yaklaşımdan söz ediliyor. Bu yaklaşımlardan biri olan ve İsveç’li bilim insanı Paul Ekman tarafından geliştirilen, FACS (Facial Action Coding System) adı verilen modelde 7 duygunun insan yüzündeki yansımaları tanımlanıyor. Mutluluk, öfke, küçümseme, tiksinme, şaşkınlık, üzüntü ve korkudan oluşan bu 7 mikro ifade (Micro Expression) bilgisayarlı görme sistemleri ve yapay zekâ ile insan gözünden daha detaylı olarak saptanabiliyor. Diğer bir ölçüm setinde ise yüz işaret noktaları (FL-Facial Landmarks) olarak adlandırılan; yüzdeki burun, kaşlar, ağızın başlangıç ve bitiş noktaları ve bunların aralarındaki mesafeler incelenerek analiz yapıyor. Neredeyse tüm teknolojinin ilgi alanına giren bu konunun bir süredir üstünde çalışılmasına rağmen henüz tam olarak güvenilir sonuçlar verdiği söylenemiyor.

Gerçekleştirdiğim projede; Python ve OpenCV kullanarak; ajanı oluşturulan datasete göre eğitmek ve gerçek zamanlı bir yüz tanıma programı ortaya çıkartmaktır. Daha sonra algılanan yüzlerin duygu analizi yapılması amaçlanmıştır.

Giriş bölümünde yüz tanıma sistemleri ve duygu analizi hakkında bilgi verilmiş ve yapılan projenin amacı açıklanmıştır. İkinci bölümde bu projenin çözümünde kullanılan teknolojiler ve yüz tanıma uygulamasının gerçekleştirildiği adımlar mevcuttur. Problemin çözümüne yönelik gerçekleştirilen sistem blok tasarımı, API arayüzleriyle ilgili bilgiler aktarılmıştır.

**1**

**BÖLÜM 2. KULLANILAN TEKNOLOJİLER**

**2.1.Materyal-Metot**

**Python;** nesneye yönelimli, yorumlamalı, birimsel, işlemesi kolay ve yüksek seviyeli dillerdendir. Yapay zeka, veri madenciliği, makine öğrenmesi ve derin öğrenme de dahil olmak üzere çok geniş bir çalışma aralığına sahiptir. Sahip olduğu bir diğer özellik ise geniş kütüphaneler ile yazılım sürecini hızlandırmasıdır. Herhangi bir veri kaybına uğramadan rahatlıkla kullanılabilir.

**OpenCV**; açık kaynak kodlu bir görüntü işleme kütüphanesidir. Bu özelliğinden dolayı algoritmalar üzerinde değişiklikler yapılarak geliştirilebilir. Yüzleri ve nesneleri algılama ve tanımlama, videolarda insan davranışlarını sınıflandırma, görüntüleri yüksek çözünürlükte birleştirmek gibi çeşitli alanlarda başarılı bir şekilde kullanılabilir.

**Keras;** neredeyse her tür derin öğrenme modelini tanımlamak ve eğitmek için uygun bir yol sağlayan Python için bir derin öğrenme kütüphanesidir. Keras; Tensorflow, Theano ve CNTK üzerinde çalışabilen Python ile yazılmış bir üst düzey sinir ağları API’sıdır. İçerdiği çok fazla işlevsel fonksiyon sayesinde Keras kolayca bir derin öğrenme modeli oluşturmamızı ve onu eğitmemizi sağlıyor. Bu nedenle derin öğrenmeye yeni başlayanlara önerilen kütüphanelerin başında Keras geliyor.

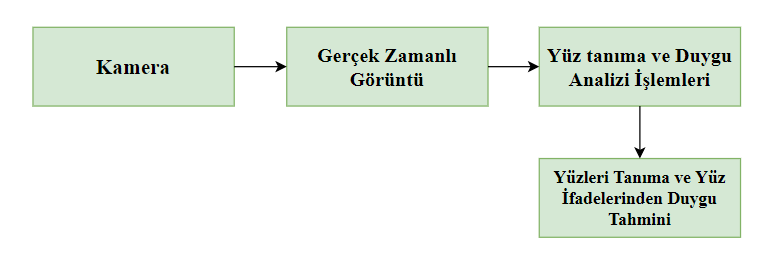
**TensorFlow;** yapay zekanın derin öğrenme çalışmaları yapan geliştiriciler için üretilmiş, açık kaynaklı bir kütüphanedir. Tek bir platforma hizmet eden bir kütüphane değildir. Çok çeşitli alanlar için bilgi sağlayarak geliştiricilerin kullanımına sunan bir frameworktür. Sahip olduğu esnek yapısıyla herhangi bir platformda, tek bir API ile bir veya birden fazla CPU, GPU kullanarak konuşlandırmanıza, yani deploy etmenize olanak da sağlıyor.

Projeyi gerçekleştirirken PyCharm ve Anaconda’dan yararlandım. Anaconda data science için gereken farklı gereçleri kapsayan bir dağıtım. İçinde IDE 'ler, editör, Python ve R için interpretör var. DS odaklı paketler hazır yüklenmiş şekilde geliyor ve kendi pip alternatifi paket yöneticisi var. İçerisinde birçok farklı kütüphaneyi barındırıyor. Örneğin; makine öğrenmesi kütüphaneleri, görüntü işleme kütüphaneleri, veri işleme kütüphaneleri gibi. Makine öğrenimi için Python’ı oldukça kolaylaştırıyor. Ek olarak TensorFlow ve OpenCv gibi çeşitli kütüphanelerden de yararlandım.

**2**

**2.2.Sistemin Genel Yapısı**

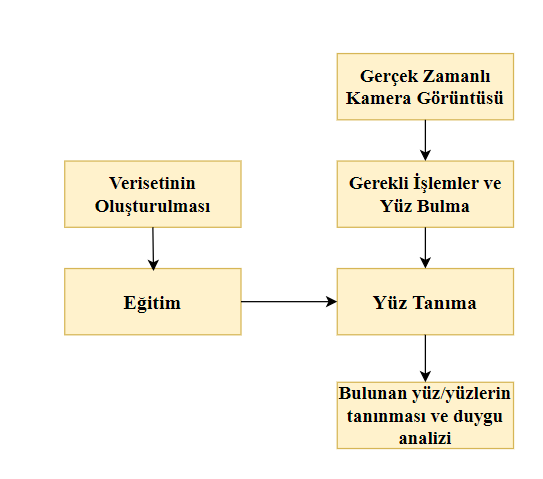
Projede; oluşturulan veri setine göre eğitilmiş programın, gerçek zamanlı kamera görüntüsündeki yüzleri tanımasını sağlayan ve tanınan yüzlerin ifadelerinden kişinin duygu tahminini gerçekleştiren bir sistem oluşturulmuştur. Buradaki problemim, gerçek zamanlı yüz görüntülerinin program tarafından algılanabilmesini sağlayabilmek ve duygu analizini doğru bir şekilde gerçekleştirebilmektir. Problemimin çözüm algoritması aşağıdaki şekilde belirtilmiştir.(Şekil 1)



**Şekil 1. Problemin Çözüm Algoritması**

Yüz tanıma ve duygu analizi için OpenCV kütüphanesi, Keras kütüphanesi ve face recognition modüllerinden yaralanılmıştır. HaarCascade ile oluşturulan veriseti ve bu verisetiyle oluşturulan eğitim klasöründeki verilerle; kamerayla gerçek zamanlı olarak yakalanan yüzler karşılaştırılarak kişinin uyumluluğu ve adına göre yüz tanıma gerçekleştirmektedir. Verisetinde olmayan ve eğitime katılmamış yüzler kamerada tanınmamaktadır ve bu kişiler için program ‘Bilinmiyor’ tanımlaması yapmaktadır. Duygu analizi için ise ‘Kızgın’, ‘Mutlu’, ‘Nötr’, ‘Üzgün’, ‘Şaşkın’ olarak programa öğretilen insan duygularını, yüz ifadelerinin analizini yaparak duygu analizi gerçekleşmektedir. Aşağıda sistemin genel yapısı verilmiştir.

(Şekil 2)



**Şekil 2. Sistemin Genel Yapısı**

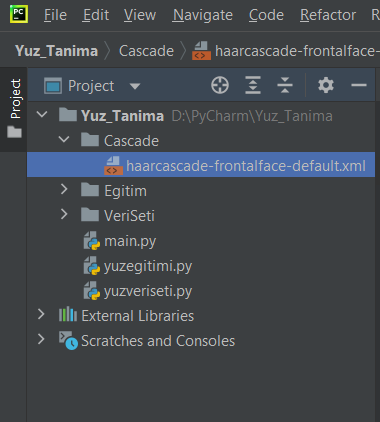
**3**

**BÖLÜM 3.YÜZ TANIMA UYGULAMASI**

Bu projesi 3 ayrı kısımdan oluşuyor. İlk kısım, veri setini oluşturmamızı sağlıyor. İkinci kısım veri setini eğitiyor. Üçüncü kısım ise bilgisayarın kamerası aracılığıyla gerçek zamanlı aldığı görüntü ile yüz tanıma işlemlerini canlı olarak gerçekleştiriyor.

**3.1. Veri Seti Oluşturma**

Veri seti oluşturmada; yüzlerin algılanabilmesi için önceden veri kümesi oluşturuyoruz. Öncelikle yüzleri algılayabilmesi için bir Cascade klasörü açtım ve içine ‘haarcascade\_frontalface\_default.xml’ classifier’ını kaydettim. (Şekil 3)

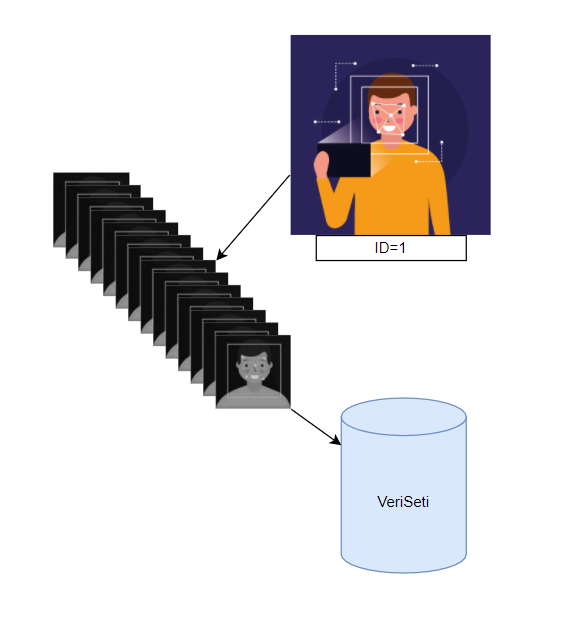


**Şekil 3.Cascade Klasörü**

Daha sonra veri setini oluşturacak kişi fotoğraflarının kaydedilmesi için bir ‘VeriSeti’ klasörü oluşturdum. Yakalanan her bir karenin VeriSeti klasörüne kaydedilmesini sağladım. Her bir kişi için farklı ID numaraları girilmesini istedim ve her kişiden maksimum 50 örnek fotoğraf aldım. Maksimum fotoğraf sayısını istersek değiştirebiliriz. Ne kadar çok örnek veri o kadar iyi sonuç elde etmemizi sağlar. Ancak fotoğraf sayısı arttıkça, veri seti oluşturma ve eğitimde zaman olarak da bir artık gözlemlenecektir. Yeni bir yüz eklemek istediğimizde ya da ID numarasını değiştirmek istediğimizde tekrardan ‘yuzveriseti.py’ çalıştırmak yeterli olaraktır. Çünkü program sonlandığında bellek temizleniyor.

**4**

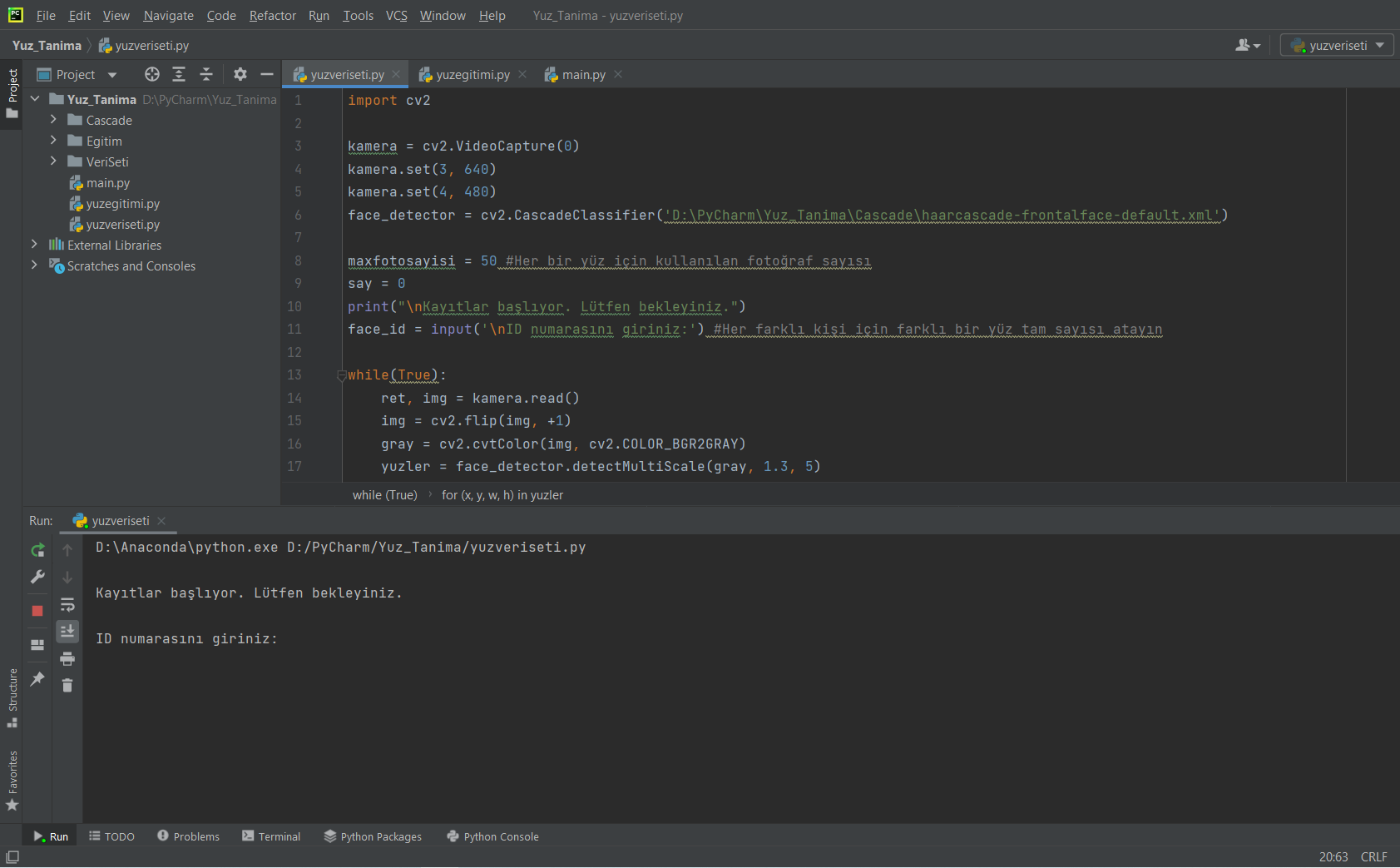
Veri setine kaydedilen yüzler, anlık olarak kameradan yakalanıyor ve kaydediliyor. (Örnek olarak Şekil 4)



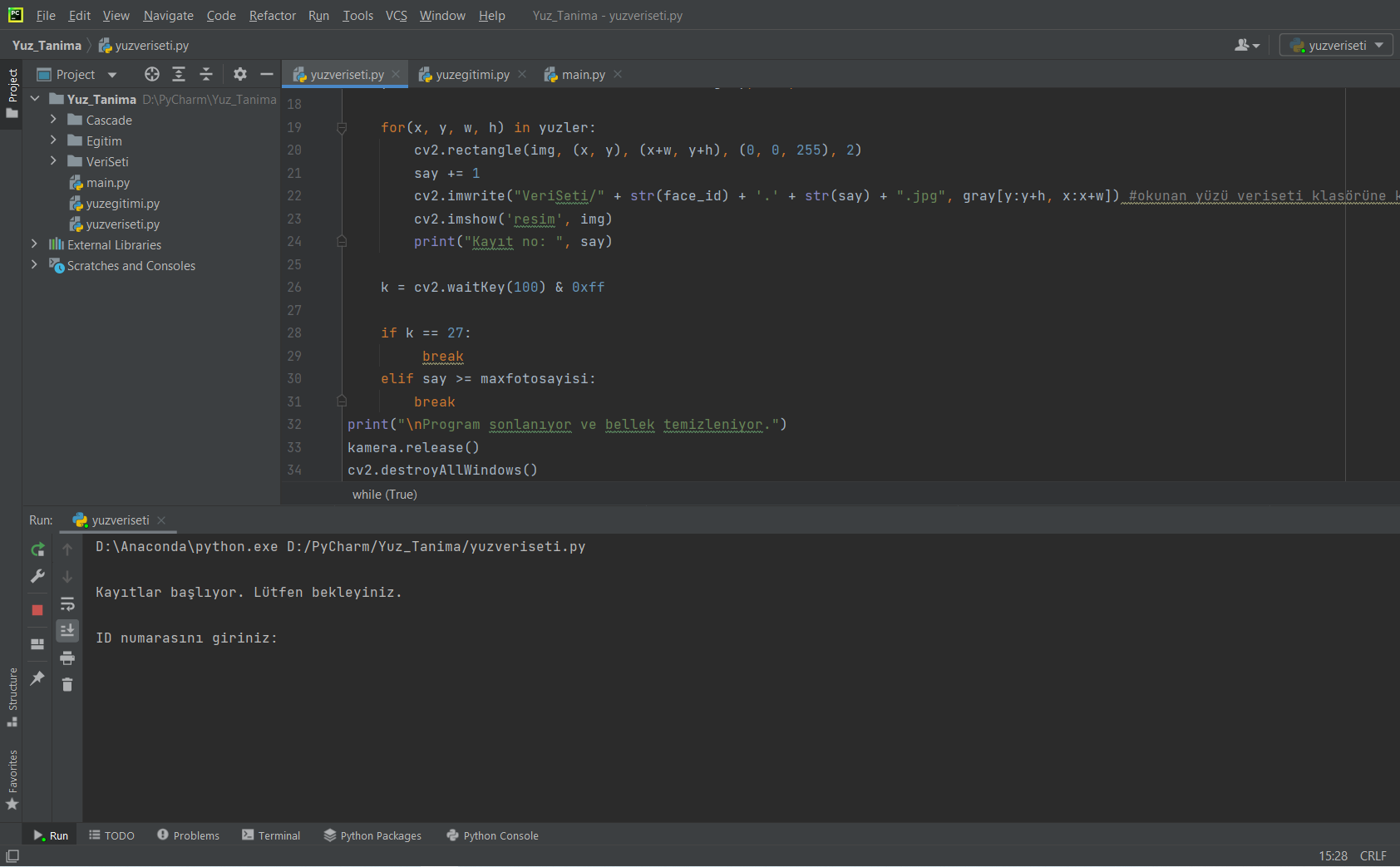
**Şekil 4. Yüzlerin Veri Setine Kaydedilmesi**

Veri seti oluşturma kodları ve çalıştırdıktan sonraki çıktıları ekran kayıtları şeklinde aşağıda

gösterilmiştir.(Şekil 5.1, Şekil 5.2, Şekil 6, Şekil 7, Şekil 8)

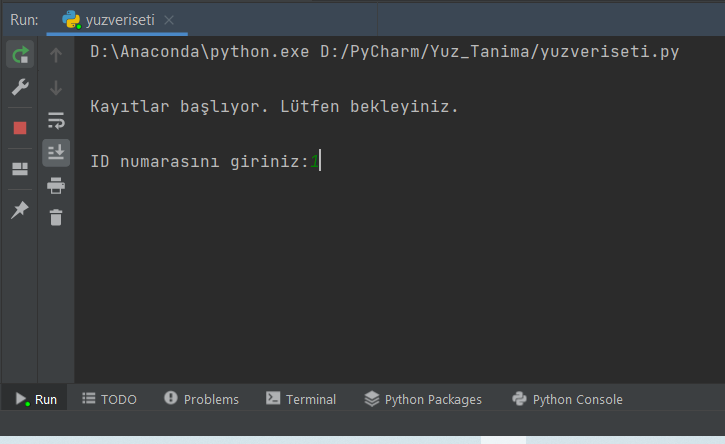


**Şekil 5.1. Yüz Veri Seti Oluşturma Kodları**

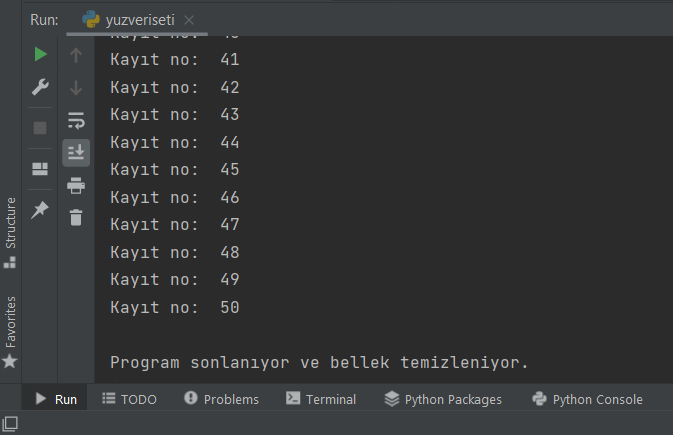


**Şekil 5.2. Yüz Veri Seti Oluşturma Kodları**

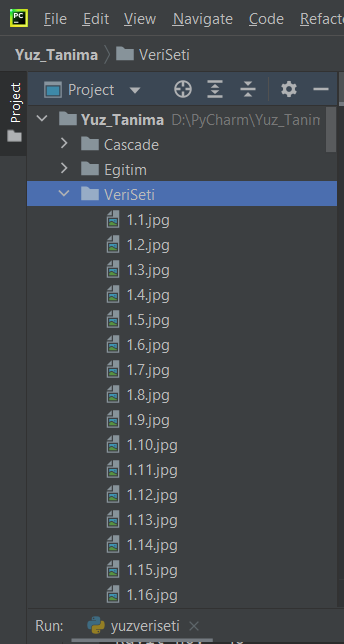
**5**



**Şekil 6. Yüzün ID numarası Girişi**



**Şekil 7. Kaydedilen Görüntüler**

****

**Şekil 8. İşlemden Sonra Veri Seti**

**6**

yuzveriseti’nin kodları aşağıdaki gibidir;

import cv2

kamera = cv2.VideoCapture(0)

kamera.set(3, 640)

kamera.set(4, 480)

face\_detector = cv2.CascadeClassifier('D:\PyCharm\Yuz\_Tanima\Cascade\haarcascade-frontalface-default.xml')

maxfotosayisi = 50 #Her bir yüz için kullanılan fotoğraf sayısı

say = 0

print("\nKayıtlar başlıyor. Lütfen bekleyiniz.")

face\_id = input('\nID numarasını giriniz:') #Her farklı kişi için farklı bir yüz tam sayısı atayın

while(True):

ret, img = kamera.read()

img = cv2.flip(img, +1)

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

yuzler = face\_detector.detectMultiScale(gray, 1.3, 5)

for(x, y, w, h) in yuzler:

cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 255), 2)

say += 1

cv2.imwrite("VeriSeti/" + str(face\_id) + '.' + str(say) + ".jpg", gray[y:y+h, x:x+w])

cv2.imshow('resim', img)

print("Kayıt no: ", say)

k = cv2.waitKey(100) & 0xff

if k == 27:

break

elif say >= maxfotosayisi:

break

print("\nProgram sonlanıyor ve bellek temizleniyor.")

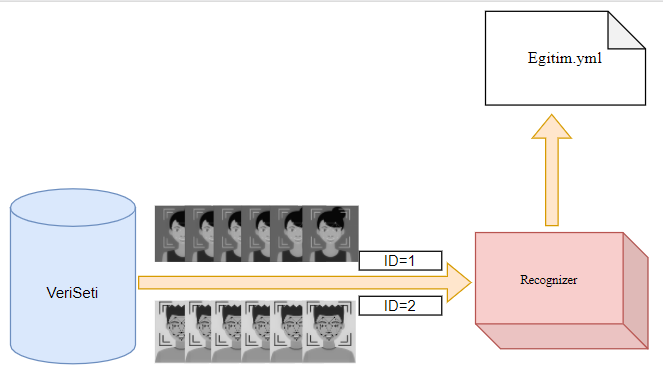
kamera.release()

cv2.destroyAllWindows()

**7**

**3.2. Eğitim**

Eğitim kısmında; OpenCV Recognizer’ı eğitici olarak kullandım ve bütün kullanıcılar verilerini oluşturduğum veri setinden almasını sağladım. (Şekil 9)

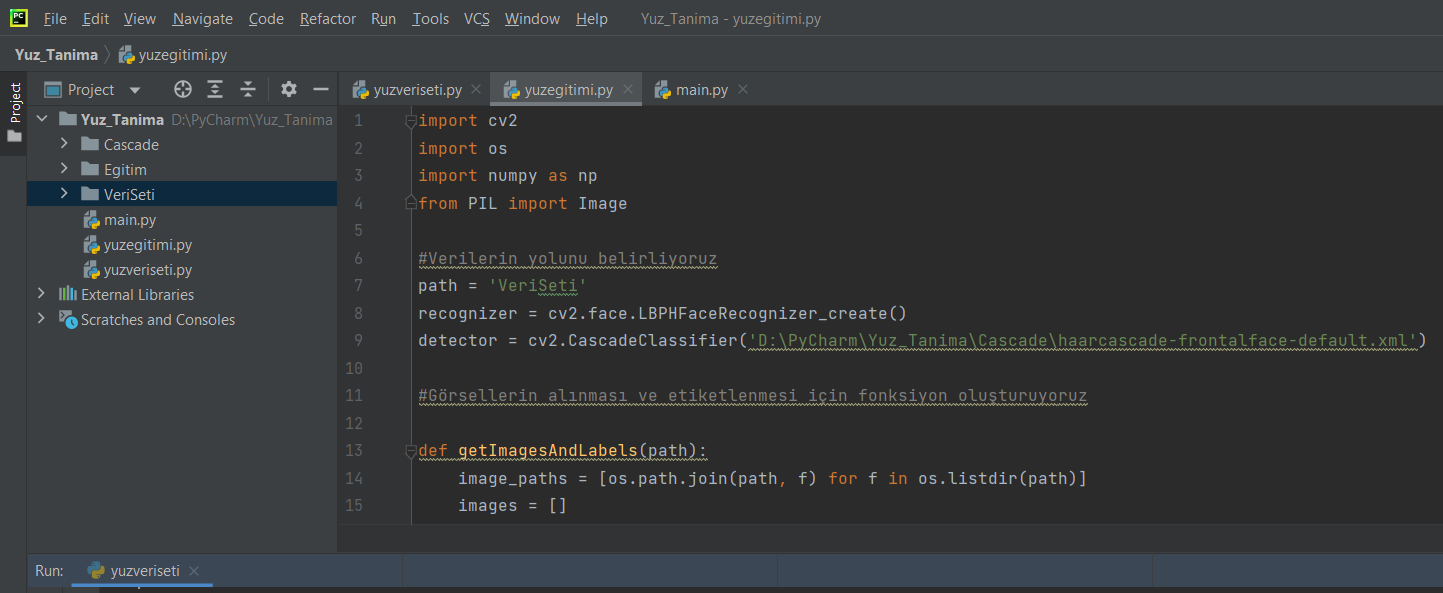


**Şekil 9. Eğitim Süreci**

Recognizer olarak OpenCV paketinde bulunan LBPH yüz tanımayı kullandım. Oluşturduğum ‘getImagesAndLabels(path)’ fonksiyonu dizindeki fotoğrafları çekti ve veri setine 2 dizi olarak döndürdü. Bunlar yuzler ve labels. Ve bu dizilerle Recognizer’ ı eğittim. Eğitim sonucunda, önceden oluşturduğum ‘Egitim’ klasörü ‘Egitim.yml’ dosyasını içeriyordu.

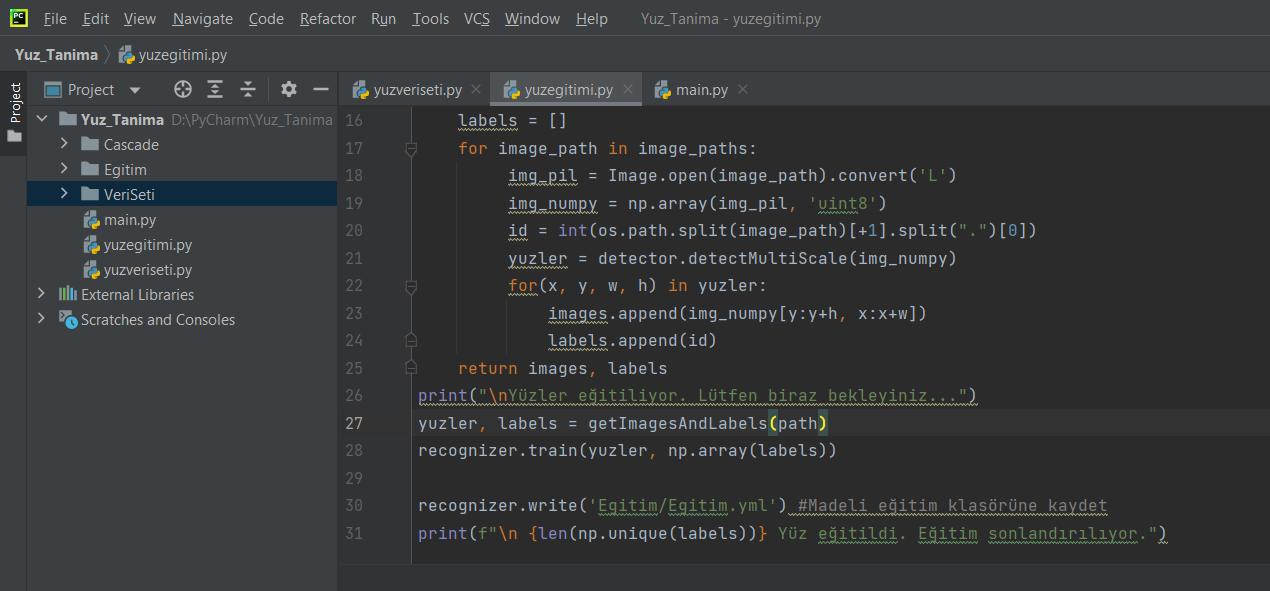
Eğitim kodları ve çalıştırdıktan sonraki çıktıları ekran kayıtları şeklinde aşağıda gösterilmiştir.

(Şekil 9.1, Şekil 9.2, Şekil 10, Şekil 11)

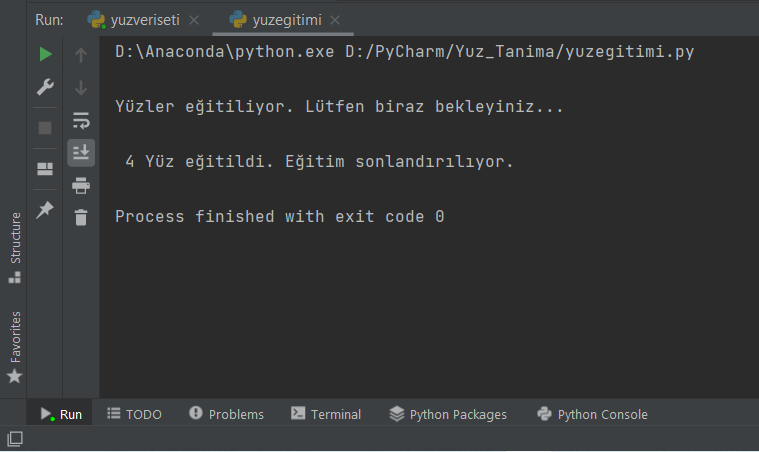


**Şekil 9.1. Eğitim Kodları**

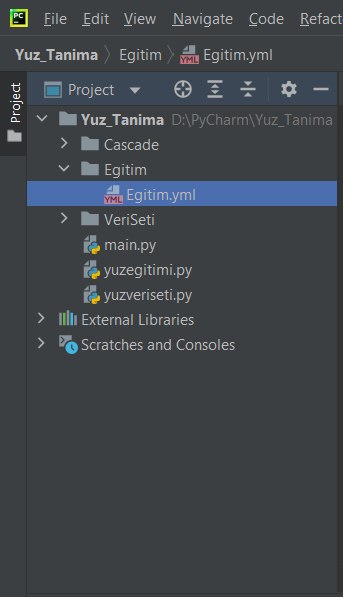
**8**

****

**Şekil 9.2. Eğitim Kodları**

****

**Şekil 10. Eğitim Sonucu Çıktı**

****

**Şekil 11. Egitim Klasöründe Oluşan ‘Egitim.yml’ Dosyası**

**9**

yuzegitimi’in kodları aşağıdaki gibidir;

import cv2

import os

import numpy as np

from PIL import Image

#Verilerin yolunu belirliyoruz

path = 'VeriSeti'

recognizer = cv2.face.LBPHFaceRecognizer\_create()

detector = cv2.CascadeClassifier('D:\PyCharm\Yuz\_Tanima\Cascade\haarcascade-frontalface-default.xml')

#Görsellerin alınması ve etiketlenmesi için fonksiyon oluşturuyoruz

def getImagesAndLabels(path):

image\_paths = [os.path.join(path, f) for f in os.listdir(path)]

images = []

labels = []

for image\_path in image\_paths:

img\_pil = Image.open(image\_path).convert('L')

img\_numpy = np.array(img\_pil, 'uint8')

id = int(os.path.split(image\_path)[+1].split(".")[0])

yuzler = detector.detectMultiScale(img\_numpy)

for(x, y, w, h) in yuzler:

images.append(img\_numpy[y:y+h, x:x+w])

labels.append(id)

return images, labels

print("\nYüzler eğitiliyor. Lütfen biraz bekleyiniz...")

yuzler, labels = getImagesAndLabels(path)

recognizer.train(yuzler, np.array(labels))

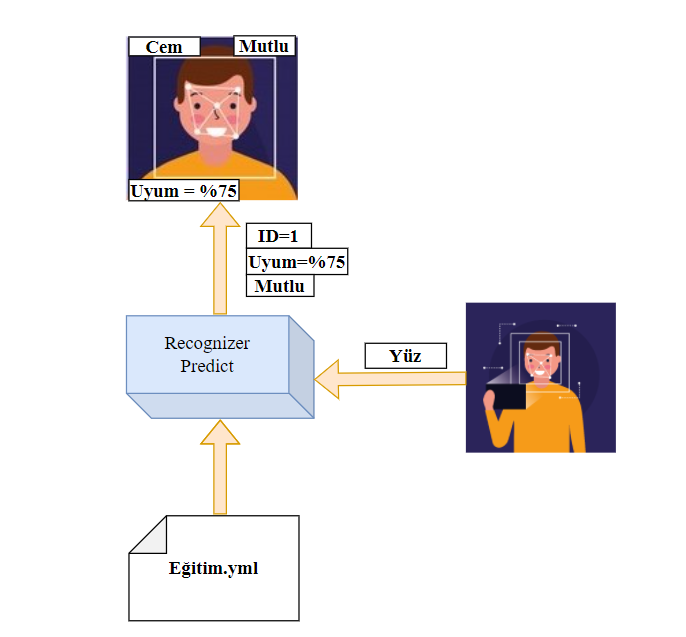
recognizer.write('Egitim/Egitim.yml') #Madeli eğitim klasörüne kaydet

print(f"\n {len(np.unique(labels))} Yüz eğitildi. Eğitim sonlandırılıyor.")

**10**

**3.3. Yüz Tanıma ve Duygu Analizi**

Projenin son kısmı olan yüz tanıma ve duygu analizi kısmında; kamera gerçek zamanlı olarak karşısındaki yüzü yakalayacak. Eğer bu kişinin yüzü daha önce veri setinde bulunup, eğitilmişse; kişinin ID numarasını döndürerek kim olduğu ve uyumluluğunun ne kadar olduğu hakkında bilgi verecek. Aynı zamanda kişinin yüzündeki ifadeye göre duygu analizi yaparak kişinin, ‘Kızgın’, ‘Mutlu’, ‘Nötr’, ‘Üzgün’, ‘Şaşkın’ ifadelerinden hangisine sahip olduğunu tahmin edecek. (Şekil 12)



**Şekil 12. Yüz Tanıma ve Duygu Analizi**

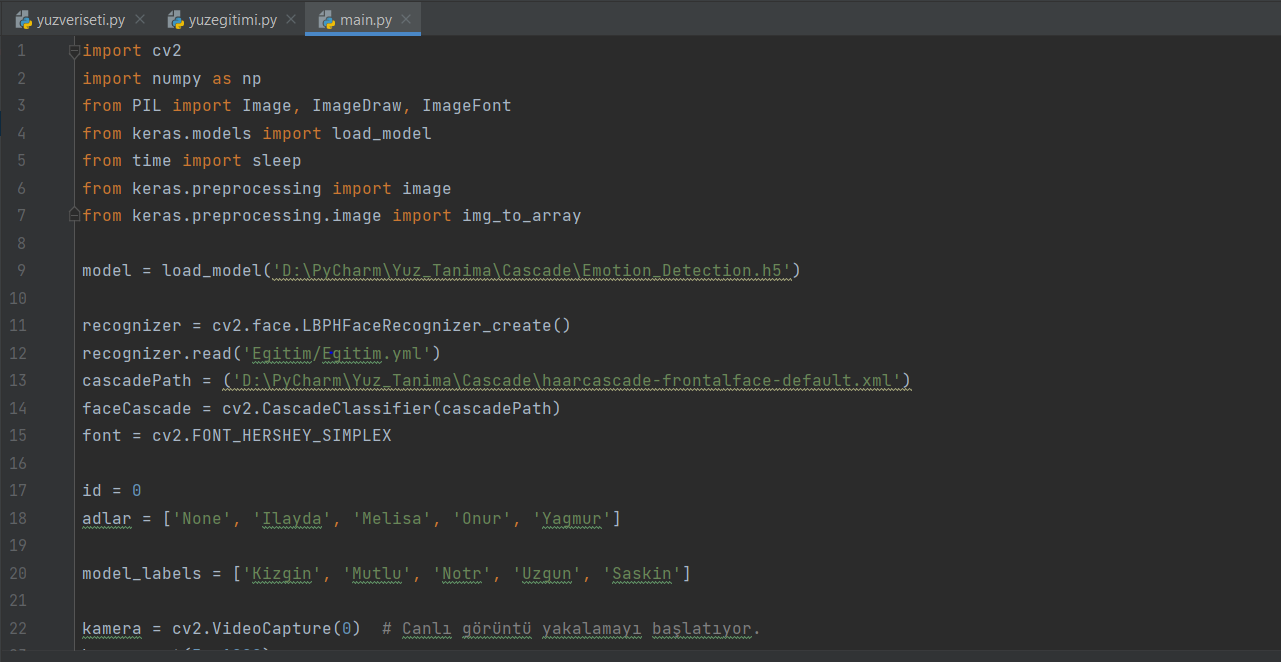
ID numaraları yerine isimlerin gözükmesi için ‘adlar’ isminde bir dizi oluşturdum. Örneğin ID=1 olan kişi İlayda, ID=2 olan kişi Melisa vb.

Ardından daha önce de yaptığım gibi HaarCascade ile yüz tespit etme işlemini uyguladım. Recognizer.predict() ile yakalanan yüzü parametre olarak aldım ve kimliğini eşleştirmeye çalıştım. Eğer ki yüzü tahmin edebiliyorsa; görüntünün üst kısmında olası adını ve alt kısmına da uyumluluğunu yazdırdım. Yakaladığı yüzü tahmin edemiyorsa da görüntünün üst kısmında ‘Bilinmiyor’ yazdırdım. Duygu analizi için erişimsel sinir ağlarından yardım aldım. Erişimsel sinir ağları görüntü tanıma ve doğal dil işleme konularına dayalı sınıflandırmalarda kullanılıyor. Duygu sınıflandırma işlemiyle kişinin duygu analizini gerçekleştirdim.

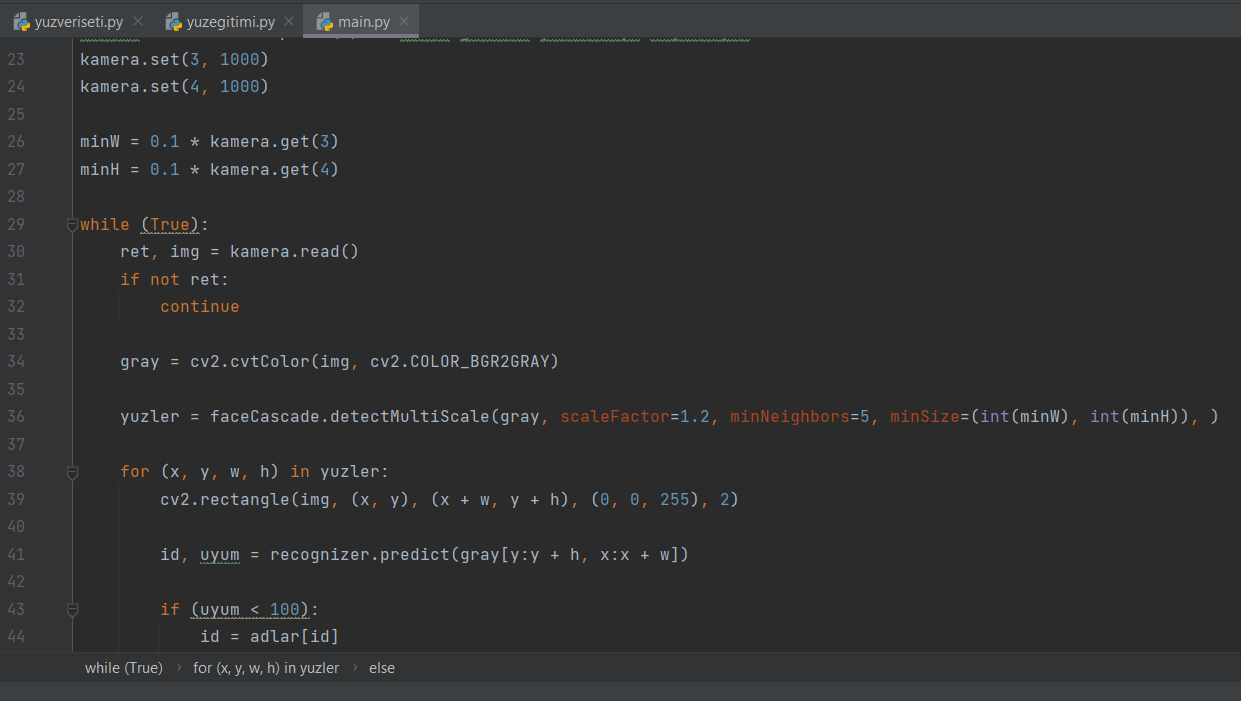
**11**

Yüz Tanıma kodları ve çalıştırdıktan sonraki çıktıları ekran kayıtları şeklinde aşağıda

gösterilmiştir. (Şekil 13.1, Şekil 13.2, Şekil 13.3, Şekil13.4, Şekil 14.1, Şekil 14.2, Şekil 14.3)

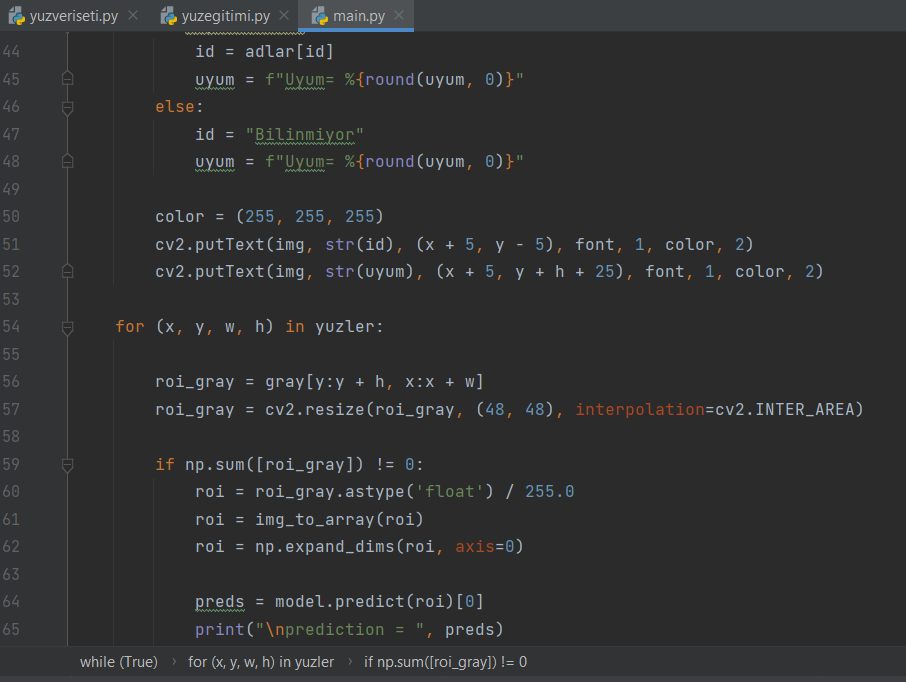


**Şekil 13.1. Yüz Tanıma ve Duygu Analizi Kodları**

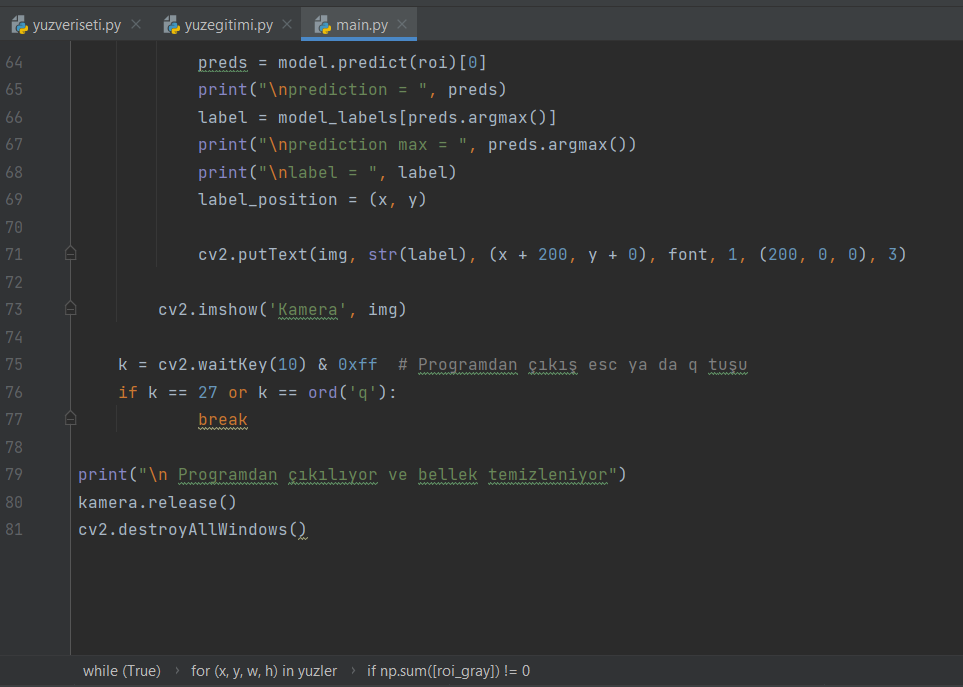


**Şekil 13.2. Yüz Tanıma ve Duygu Analizi Kodları**

**12**

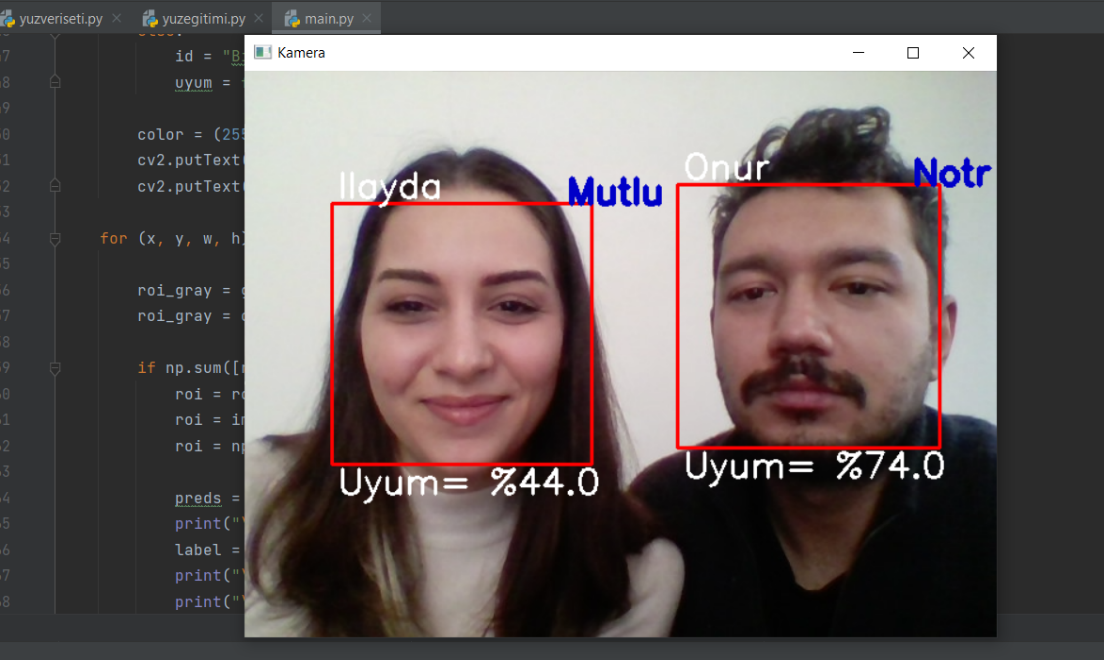


**Şekil 13.3. Yüz Tanıma ve Duygu Analizi Kodları**

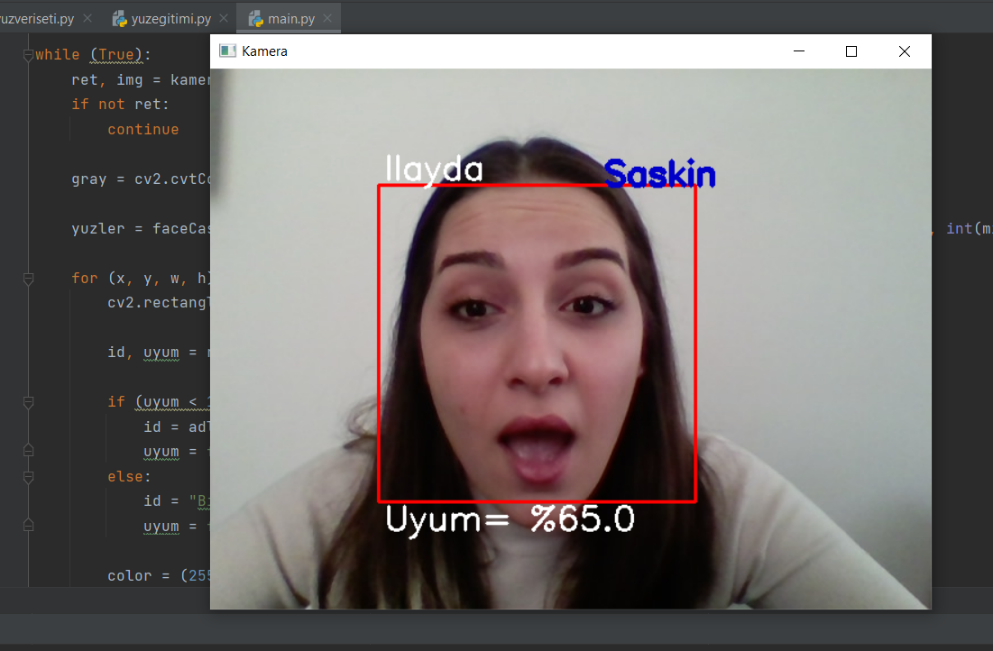


**Şekil 13.4. Yüz Tanıma ve Duygu Analizi Kodları**

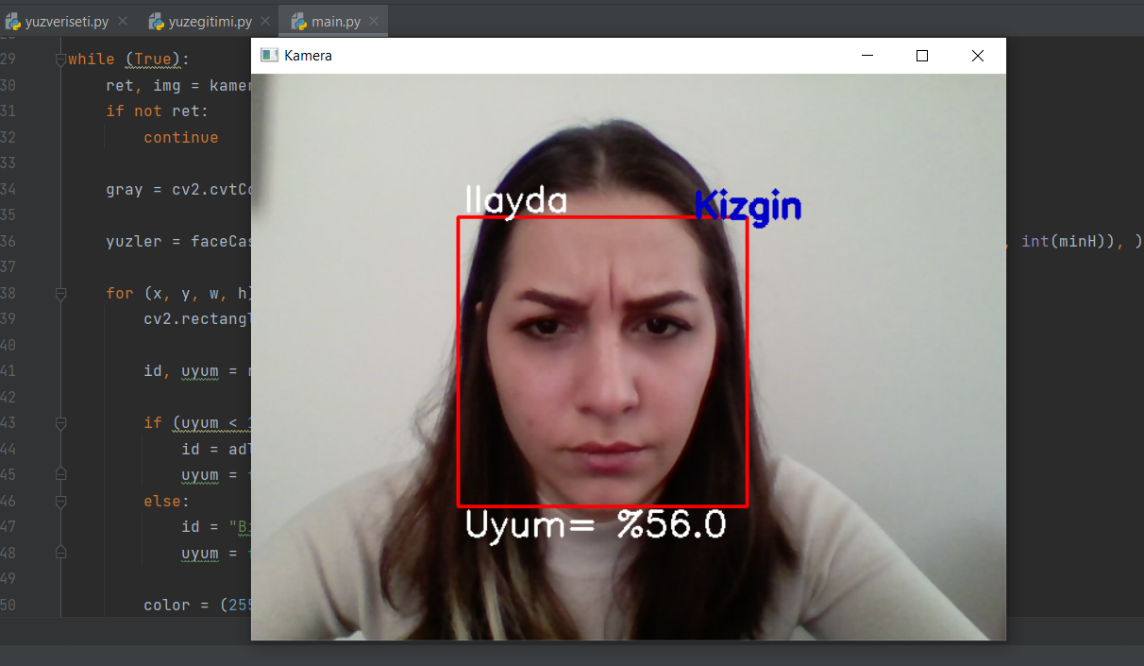
**13**



**Şekil 14.1. Program çıktısı**



**Şekil 14.2. Program çıktısı**

****

**Şekil 14.3. Program çıktısı**

**14**

main’ in kodları aşağıdaki gibidir;

import cv2

import numpy as np

from PIL import Image, ImageDraw, ImageFont

from keras.models import load\_model

from time import sleep

from keras.preprocessing import image

from keras.preprocessing.image import img\_to\_array

model = load\_model('D:\PyCharm\Yuz\_Tanima\Cascade\Emotion\_Detection.h5')

recognizer = cv2.face.LBPHFaceRecognizer\_create()

recognizer.read('Egitim/Egitim.yml')

cascadePath = ('D:\PyCharm\Yuz\_Tanima\Cascade\haarcascade-frontalface-default.xml')

faceCascade = cv2.CascadeClassifier(cascadePath)

font = cv2.FONT\_HERSHEY\_SIMPLEX

id = 0

adlar = ['None', 'Ilayda', 'Melisa', 'Onur', 'Yagmur']

model\_labels = ['Kizgin', 'Mutlu', 'Notr', 'Uzgun', 'Saskin']

kamera = cv2.VideoCapture(0) # Canlı görüntü yakalamayı başlatıyor.

kamera.set(3, 1000)

kamera.set(4, 1000)

minW = 0.1 \* kamera.get(3)

minH = 0.1 \* kamera.get(4)

while (True):

ret, img = kamera.read()

if not ret:

continue

**15**

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

yuzler = faceCascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5, minSize=(int(minW), int(minH)), )

for (x, y, w, h) in yuzler:

cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

id, uyum = recognizer.predict(gray[y:y + h, x:x + w])

if (uyum < 100):

id = adlar[id]

uyum = f"Uyum= %{round(uyum, 0)}"

else:

id = "Bilinmiyor"

uyum = f"Uyum= %{round(uyum, 0)}"

color = (255, 255, 255)

cv2.putText(img, str(id), (x + 5, y - 5), font, 1, color, 2)

cv2.putText(img, str(uyum), (x + 5, y + h + 25), font, 1, color, 2)

for (x, y, w, h) in yuzler:

roi\_gray = gray[y:y + h, x:x + w]

roi\_gray = cv2.resize(roi\_gray, (48, 48), interpolation=cv2.INTER\_AREA)

if np.sum([roi\_gray]) != 0:

roi = roi\_gray.astype('float') / 255.0

roi = img\_to\_array(roi)

roi = np.expand\_dims(roi, axis=0)

**16**

preds = model.predict(roi)[0]

print("\nprediction = ", preds)

label = model\_labels[preds.argmax()]

print("\nprediction max = ", preds.argmax())

print("\nlabel = ", label)

label\_position = (x, y)

cv2.putText(img, str(label), (x + 200, y + 0), font, 1, (200, 0, 0), 3)

cv2.imshow('Kamera', img)

k = cv2.waitKey(10) & 0xff # Programdan çıkış esc ya da q tuşu

if k == 27 or k == ord('q'):

break

print("\n Programdan çıkılıyor ve bellek temizleniyor")

kamera.release()

cv2.destroyAllWindows()

**17**

**BÖLÜM 4. SONUÇ**

İnsan beyni dünyanın en karmaşık makinesi olarak kabul edilebilir. İnsan beyni sayısal bir işlemi birkaç dakikada yapabilmesine karşın; idrak etmeye yönelik işlemleri çok daha kısa bir sürede yapabilir. Burada “Bilgisayarların yardımı ile böyle bir zekâ üretilebilir mi?” sorusu sorulabilir. Bu olayda insanı ya da insan beynini üstün kılan temel özellik sinirsel algılayıcılar sayesinde kazanılmış ve göreli olarak sınıflandırılmış bilgileri kullanabilmesidir. İnsan beyninin bu özellikleri yapay zeka araştırmaları için ilham kaynağı olmuştur.

Bir bilgisayarın, ister bir suçlunun tespiti, ister bir oy verme işleminin otonomlaştırılması olsun; yorumlayarak tanımayı gerçekleştirebilmesi yapay zekâ araştırmacılarını ilgilendiren önemli konulardandır.

Yüz tanıma sistemlerinde birinci adım yüz bölgesinin tanınması ve arka plandan ayrılarak çıkarılmasıdır. Bu konunun, içerik tabanlı görüntü erişimi, video kodlama, video konferans, kalabalık alanların izlenmesi gibi pek çok alanda uygulaması bulunmaktadır. İnsan yüzü dinamik bir nesnedir ve değişkenliğe sahiptir. Özellikle bilgisayarlı görüntü sistemleriyle entegre edildiğinde, bu durum yüz algılamayı zor bir problem haline getirir. Bu zor problem neticesinde yüz algılamada 150'den fazla farklı yaklaşım ve araştırma bulunmaktadır. Yüz tanıma sistemleri ve fotoğrafları çözümleyerek etiketleyen sistemler veri sayısı artıkça yapay zekâ sistemleri sayesinde son derece başarılı sonuçlar elde etmektedir. Yine bu sistemler duyguları analiz etmek için de kullanılabilmektedir.

Yüz ifadesi gibi psikolojik durumun değerlendirilmesinin, insanlar tarafından bile bazen kolay olmadığı düşünüldüğünde, bilgisayar ile bu işlemin gerçekleşmesinin ne derecede zor olduğu anlaşılmaktadır. Yüz ifadesinin bilgisayar ile tanınması resimden yüz tespiti ve yüze ilişkin ifadenin değerlendirilmesi olmak üzere 2 temel aşama içermektedir. İlk aşama olan resimden yüzün bulunması, genellikle ten işleme ile iyileştirilmiş yapay sinir ağları yaklaşımı ile gerçekleştirilmektedir. Daha sonra yüze ait psikolojik durum değerlendirilmesi yapıldığında, yüz ifadesi için “ikili yerel örnekler operatörü”, sınıflama için ise örneğin “Ki kare istatistiği” ve uzaklık ölçütleri kullanılabilmektedir.

Ayrıca bu sistemler özellikle çok kıymetli materyallere sahip bilgi merkezlerinin güvenliğinin sağlanmasında çok önemli roller oynayabilirler. Kıymetli materyallere sahip kütüphaneler, arşivler, müzeler vb. pek çok bilgi merkezleri, yüz tanıma teknolojilerini güvenlik nedeniyle kullanabilir. Kameralar aracılığı ile kişinin yüzündeki ifadeden duygu analizi yapabilen sistemler de bilgi merkezlerine çeşitli fırsatlar sunmaktadır. Kullanıcılarının ve çalışanlarının, bilgi merkezinde geçirdikleri süre boyunca yüzlerinde oluşan duygu ifadeleri, yapay zekâ sistemleri tarafından yakalanabilir.

Bu sistemleri kullanırken konunun etik boyutu da unutulmamalı, etik ihlallerden kaçınılmalı ve bu

sistemleri kullanmak isteyen bilgi merkezleri tarafından ilgili konu ile ilgili politikalar oluşturulmalıdır.

**18**