

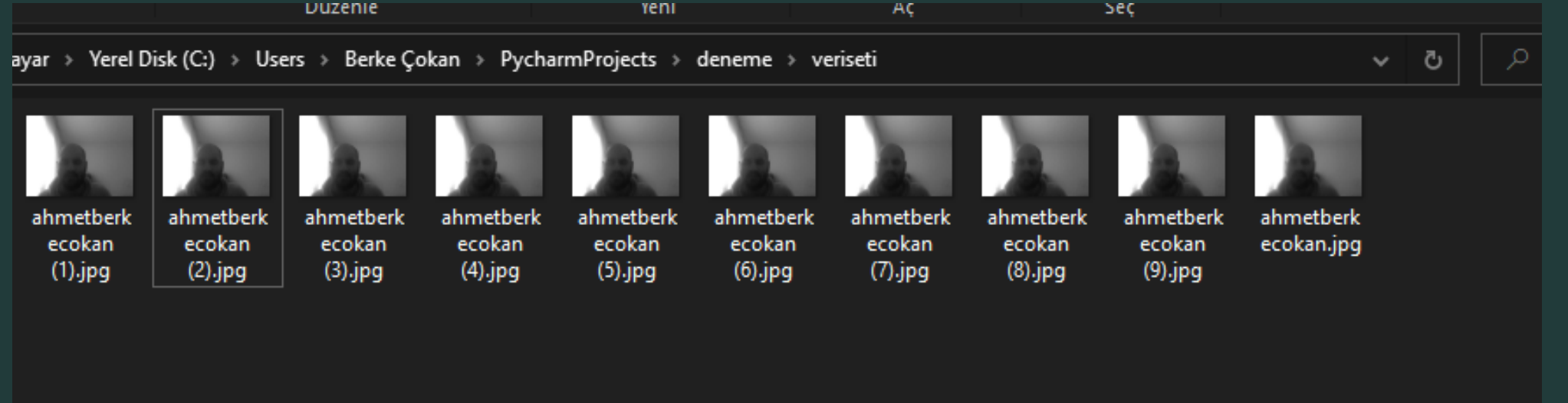
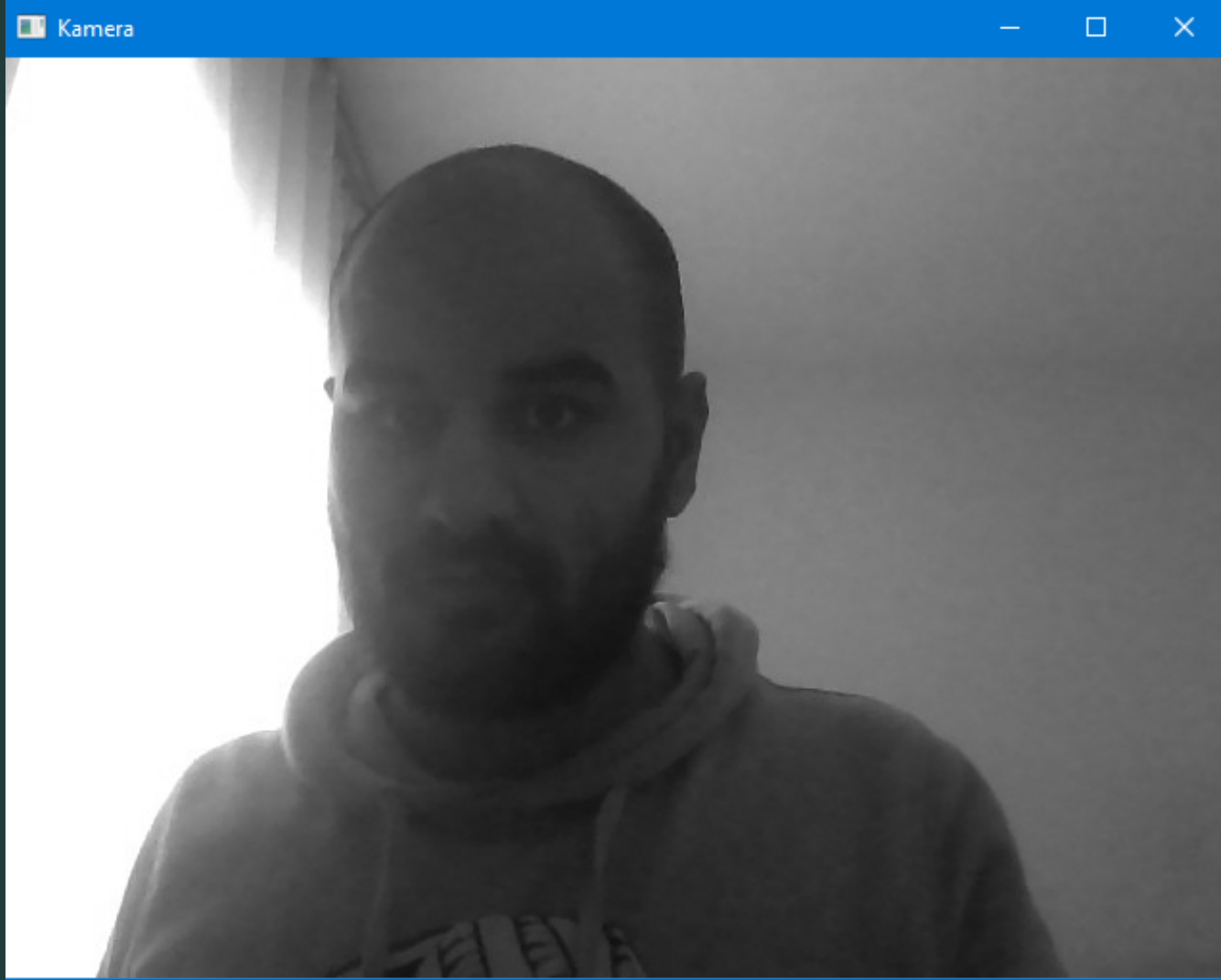
PROJEM



- Stajımın ilk haftasında sorumlu mühendisim benden fabrikada gözlem yapmamı ve sorunları tespit etmemi istedi. Bunun sonucunda bende fabrikadaki taşeron elemanlarının yemekhaneden yararlanmadığı halde yemek yediklerini, bunun önüne geçilebilmesi için güvenlik personelinin orada denetim yaptığını bunu bir yüz tanıma uygulaması ile çözülebileceğini belirttim ve proje olarak buna karar verildi.

- Projede Python programlama dilini kullandım. Projemde modeli eğitebilmek için fabrikada çalışıp yemek hakkı olan personelin yüzlerinin sisteme atılması lazımdı. Bunun için projemizde kamerayı açıyoruz. Kameramız açıldıktan sonra öncelikle datasete atılacak personelin adını sisteme girmemiz gerekiyor. Daha sonra 's' tuşuna basıldıktan sonra basıldığı saniye içinde 10 adet ekran görüntüsü alıp projemizin datasetine atıyor.

```
yuzkaydetme (1) ×  
"C:\Users\Berke Çokan\PycharmProjects\deneme\venv\Scripts\python.exe" "C:/Users/Berke Çokan/PycharmProjects/deneme/yuzkaydetme.py"  
Kişi adı ve soyadını giriniz Ahmet berke Cokan
```



PROJEMİN BU KISMININ KODU

```
import cv2 as cv

kisismi=input("Kişi adı ve soyadını giriniz")

kayit = cv.VideoCapture(0)
kisisayisi=0

while kayit.isOpened():
    ret, frame = kayit.read()
    griTon = cv.cvtColor(frame, cv.COLOR_RGB2GRAY)
    if not ret:
        print("Kamera okunamadı...")
        break
    cv.imshow("Kamera", griTon)
    tuş = cv.waitKey(1)
    if tuş == ord('q'):
        break
    elif tuş == ord('s'):
        for x in range(10):
            cv.imwrite("veriseti/"+str(kisisayisi)+".jpg", griTon)
            kisisayisi=kisisayisi+1

kayit.release()
cv.destroyAllWindows()
```

Model Eğitim Kısım

- Datasetten çektiğimiz görüntüleri kullanarak modelimiz eğitime başlıyoruz. Datasetimizin yolunu sisteme gösterdikten sonra yüzleri tanınması için "haarcascade_frontalface_default.xml" dosyasının yolunu gösteriyoruz. Bu bizim yüzleri tanımamız için gerekli olan bir xml dosyası. Daha sonra bu verilerimizi dizilere yüklüyor ve eğitimi başlatıyoruz. Sonunda "egitim.yml" dosyasını projemizin dizinine kaydediyor ve eğitimi sonlandırıyoruz.

```
yuzler,ids = getImagesAndLabels(path)
recognizer.train(yuzler, np.array(ids))
recognizer.write('egitim/egitim.yml')
print(f"\n [INFO] {len(np.unique(ids))} yüz eğitildi.")
```

Model Kodumun Tamamı

```
import numpy as np
from PIL import Image
import os

path = 'veriseti'
recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("Cascades/haarcascade_frontalface_default.xml")

def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    ornekler=[]
    ids = []
    for imagePath in imagePaths:
        PIL_img = Image.open(imagePath).convert('L')
        img_numpy = np.array(PIL_img, 'uint8')
        id = int(os.path.splitext(imagePath)[-1].split(".")[0])

        yuzler = detector.detectMultiScale(img_numpy)
        for (x,y,w,h) in yuzler:
            ornekler.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return ornekler,ids

print("\n |yuzler eğitiliyor. Birkaç saniye bekleyin ...")
yuzler,ids = getImagesAndLabels(path)
recognizer.train(yuzler, np.array(ids))
recognizer.write('egitim/egitim.yml')
print(f"\n [INFO] {len(np.unique(ids))} yüz eğitildi.")
```

- Projemizde modeli eğittik ve artık ön yüz tarafına geldik. Öncelikle modelimiz ve xml dosyamızı sisteme tanıttık. Daha sonra ise kamera ayarlarını gerçekleştirdik.

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('egitim/egitim.yml')
cascadePath = "Cascades/haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX

kamera = cv2.VideoCapture(0)
kamera.set(3, 1000)
kamera.set(4, 800)
minW = 0.1 * kamera.get(3)
minH = 0.1 * kamera.get(4)
while True:
    ret, img = kamera.read()
    gri = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

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

- Kameramızı açtık ve eğer kameradaki anlık görüntüyü modelimiz tanır ise çerçeve için alıp afiyet olsun yazmasını ayarladık.

```
while True:
    ret, img = kamera.read()
    gri = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    yuzler = faceCascade.detectMultiScale(
        gri,
        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, 255, 0), 2)
        cv2.putText(img, "Afiyet Olsun", (x, y + h + 20), cv2.FONT_HERSHEY_DUPLEX, .5, (0, 255, 0))
        id, uyum = recognizer.predict(gri[y:y + h, x:x + w])

        if (uyum < 100):
            id = id.isim
            uyum = f"Uyum= {round(uyum,0)}%"
        else:
            id = "bilinmiyor"
            uyum = f"Uyum= {round(uyum,0)}%"

    color = (255, 255, 255)
```



```

import cv2
import numpy as np
from PIL import Image, ImageDraw, ImageFont

def print_utf8_text(image, xy, text, color):
    fontName = 'FreeSerif.ttf'
    font = ImageFont.truetype(fontName, 24)
    img_pil = Image.fromarray(image)
    draw = ImageDraw.Draw(img_pil)
    draw.text((xy[0], xy[1]), text, font=font,
              fill=(color[0], color[1], color[2], 0))
    image = np.array(img_pil)
    return image

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('egitim/egitim.yml')
cascadePath = "Cascades/haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);
font = cv2.FONT_HERSHEY_SIMPLEX

```

```

kamera = cv2.VideoCapture(0)
kamera.set(3, 1000)
kamera.set(4, 800)
minW = 0.1 * kamera.get(3)
minH = 0.1 * kamera.get(4)
while True:
    ret, img = kamera.read()
    gri = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    yuzler = faceCascade.detectMultiScale(
        gri,
        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, 255, 0), 2)
        cv2.putText(img, "Afiyet Olsun", (x, y + h + 20), cv2.FONT_HERSHEY_SIMPLEX, .5, (0, 255, 0))
        id, uyum = recognizer.predict(gri[y:y + h, x:x + w])

        if (uyum < 100):
            id = "Ahmet"
            uyum = f"Uyum= {round(uyum,0)}%"
        else:
            id = "bilinmiyor"

```

```
color = (255,255,255)

cv2.imshow('kamera', img)
k = cv2.waitKey(10) & 0xff
if k == 27 or k==ord('q'):
    break

print("\n Programdan çıkıyor ve ortalığı temizliyorum")
kamera.release()
cv2.destroyAllWindows()
```

