Bitirme Projesi 2



Ad : Ege Erbilen

Okul No : 160255080

Bölümü : Bilgisayar Mühendisliği

Sınıf : 4.sınıf

**İÇİNDEKİLER**

Proje Amacı …………………………………………………………………….1

Proje Özeti ……………………………………………………………………..1

Proje Kapsamı …………………………………………………………………..1

Kaynaklar …………………….…………………………………………………1

Projede Kullanılan Materyaller …………………………………………………2

Projenin Gereklilikleri ….….……………………………………………………2

Proje Önerileri ...……………...…………………………………………………2

Proje Süreci ……………………………………………………………………..3

Proje Nasıl Kullanılır …...………………………………………………………3

Kaynak Kodlar ………………………………………………………………...15

**Proje Amacı**

Bunalımda olan insanların intihara ne kadar yatkın olduklarını saptamak.

**Proje Özeti**

Kullanıcının kamerasından alınan anlık görüntüler ile kişilerin yüz ifadesini saptamak ve eğer kullanıcı üzgün ise bunu veri tabanına kaydetmek kaydedilen verileri makine öğrenmesi ile işleyip analiz etmek.

**Proje Kapsamı**

Proje ruhsal olarak desteğe ihtiyaç duyan insanları kapsamaktadır.

**Proje İçeriği**

Proje kapsamında python kütüphanesi olan numpy, cv2, time, datatime, keras, pymongo ve matplotlib kütüphanelerinden faydalanılmıştır.

**Kaynaklar**

2- https://www.w3schools.com/python/python\_ml\_getting\_started.asp

3- https://www.w3schools.com/python/python\_ml\_mean\_median\_mode.asp

4- <https://www.w3schools.com/python/python_ml_standard_deviation.asp>

5- <https://www.w3schools.com/python/python_ml_percentile.asp>

6- https://www.w3schools.com/python/python\_ml\_data\_distribution.asp

7- https://www.w3schools.com/python/python\_ml\_normal\_data\_distribution.asp

8- https://www.w3schools.com/python/python\_ml\_scatterplot.asp

9- https://www.w3schools.com/python/python\_ml\_linear\_regression.asp

10- https://www.w3schools.com/python/python\_ml\_polynomial\_regression.asp

11- https://www.w3schools.com/python/python\_ml\_multiple\_regression.asp

1

12- <https://www.w3schools.com/python/python_ml_scale.asp>

13- https://www.w3schools.com/python/python\_ml\_train\_test.asp

15- https://medium.com/datarunner/matplotlibkutuphanesi-1-99087692102b

16- https://towardsdatascience.com/face-detection-recognition-and-emotion-detection-in-8-lines-of-code-b2ce32d4d5de

17- https://developers.google.com/ml-kit/vision/face-detection

18- https://machinelearningmastery.com/how-to-perform-face-detection-with-classical-and-deep-learning-methods-in-python-with-keras/

**Projede Kullanılan Materyaller**

Bilgisayar, internet, bilgisayar kamerası, Win10, Python, Anaconda, OpenCV ve Spyder projenin yapımı için gerekli kaynakların oluşmasına ve yürütülmesine yardımcı olmuştur.

**Sistemin Kullanım Alanları**

Geliştirilen bu yapı herkes tarafından kullanılabilir hastanelerde, evlerde, meydanlarda (toplum ruh halini analiz etmede) hatta ve hatta iş yerlerinde.

**Projenin Gereklilikleri**

Win10, Python 3.8.3 versiyonu, ilgi kütüphanelerin ekli olması, kamera, işlemci, ram, ekran kartı, elektrik ve kişinin verileri görmesi için ekran.

**Proje önerileri**

Proje şu anda masaüstü için kullanıma uygundur, geliştirilmiş olan program web için de yazılabilir. Bu sayede güncelleme ve geliştirme işlemleri daha kolay olur kullanıcılar gelen her güncelleme için tekrar indirme yapmak zorunda kalmaz ve kullanıcıların saklama yerlerini işgal etmemiş oluruz.

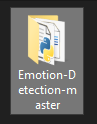
2

**Proje süreci**

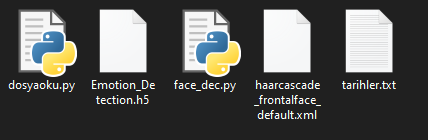
Proje süresince plan odaklı çalışma, problemleri çözebilme ve problemlere farklı bir bakış açısı ile bakma, insanların hayatlarına daha fazla nasıl umutla bakmalarını sağlaya bilirim ve onlara daha iyi nasıl yardımcı olurum konusunda bana çokça fikir verdi, pes etmemeyi ve farklı bakış açısı ile de problemlerin ortadan kaldırılabileceğini öğrendim. Proje sürecin içerisinde okulda ve okul dışında öğrendiğim bilgileri pekiştirme şansım oldu, python dili içerisinde kendimi şu anda eskiye nazaran daha hakim hissediyorum. Proje içerisinde bazen kafa karışıklığı yaşadım bu kafa karışıklığının sebebi ise 1. olarak veri tabanı için herhangi bir veri tabanı uygulamasını projesini entegre edip etmeme konusunda kararsız kaldım ve biraz araştırma yaptık dan sonra ise txt dosyası kullanmaya karar verdim bu sayede kullanıcı herhangi bir veri tabanı uygulaması kurmak zorunda kalmadı. Dosya boyutu bence hala çok yüksekti ve bu sorunu da günleri kısa tutarak çözmeye karar verdim. Dosya içerisinde ki her karakter (harfler ,sayılar, boşluk vb.) 1 bayttır, eğer dosya içerisinde alt satıra iner isek 2 bayt olur. Tabii bu bilgiden sonra verileri aynı satırda virgül kullanarak yazdırdım fakat veri txt dosyası içeriği çok karışık oldu ben de bu çözümden memnun kalmadım ve her birini alt satıra indirerek bir çözüm elde ettim. 2. olarak ise günleri kod içerisinde string yapıda 7 değişken ile uttum sonradan bu yapının karışık olduğunu düşünüp derslerde öğrendiğim tuple ile çözmeye ulaştım ve kodun anlaşılmasını kolaylaştırmış oldum.

**Proje Nasıl Kullanılır**

Öncelikle Projemizin bulunduğu “Emotion-Detection-master” klasörünü açarız.



Dosyamızı açtıktan sonra aşağıdaki gibi bir görüntü gelecektir.

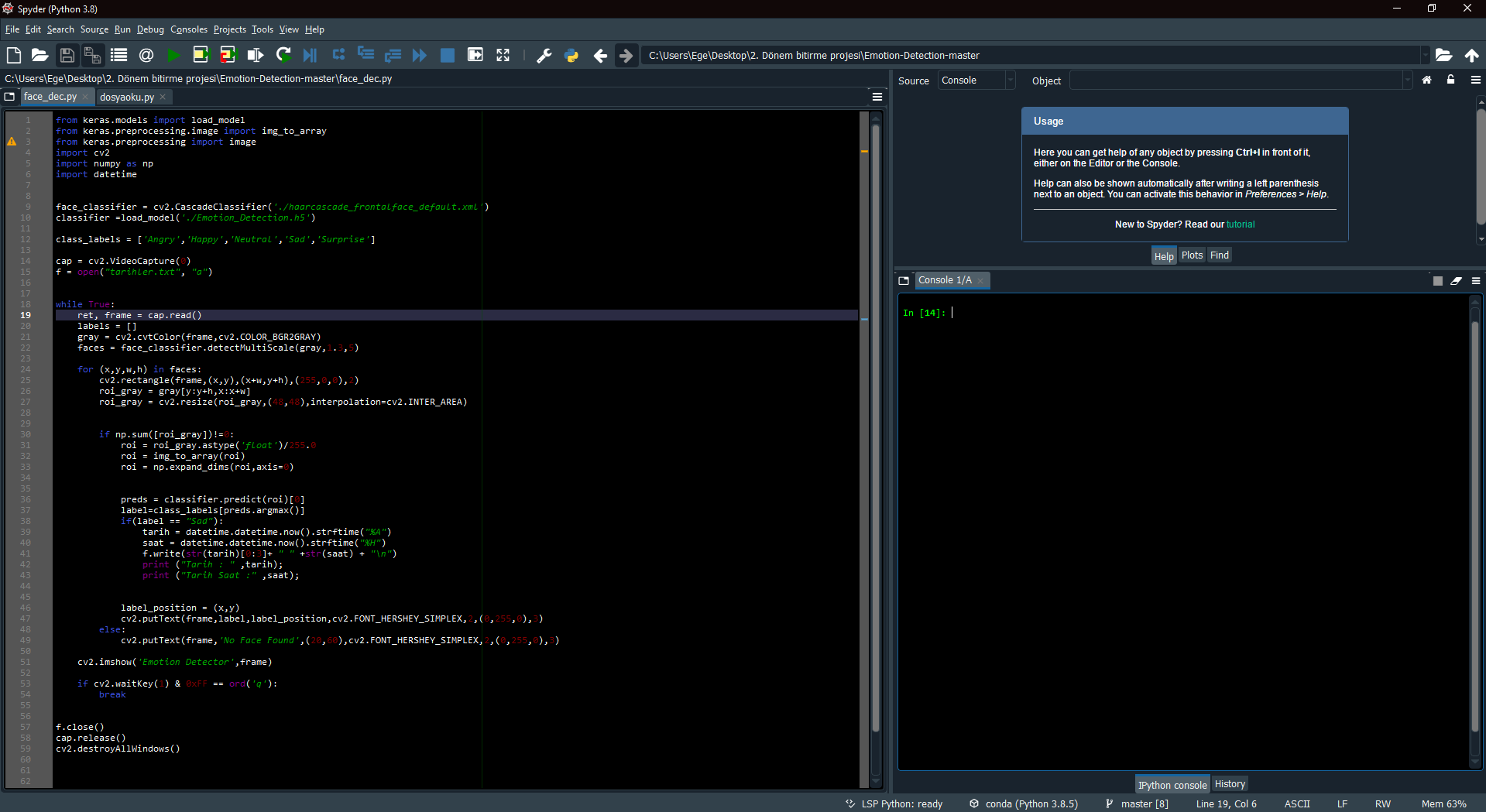


3

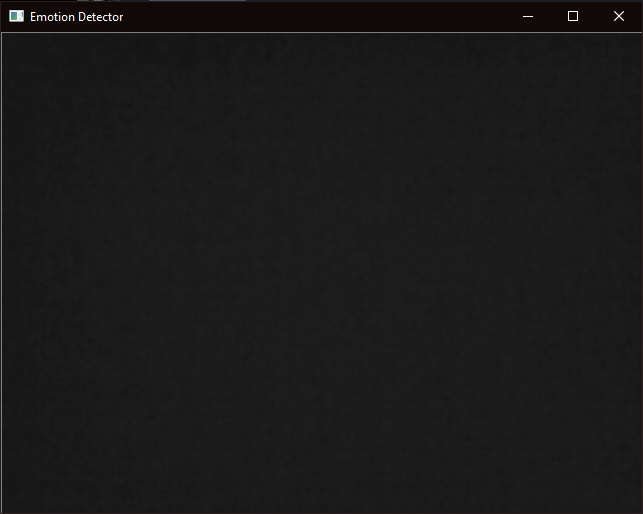
Ardından projemizi Anaconda içerisinde bulunan spyderdan açtıktan sonra

C:\Users\Ege\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Ekran Alıntısı.PNG

Burada ki “face\_dec.py” ve “dosyaoku.py” dosyasını spyder içerisine sürükle bırak yaparız ve aşağıdaki gibi bir görüntü karşımıza gelir.



4

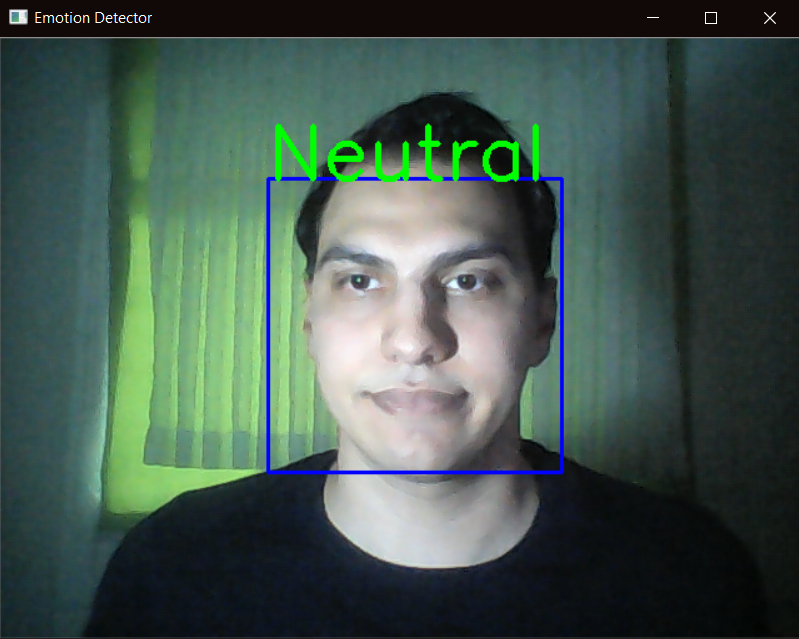
“face\_dec.py” dosyamızı açtıktan sonra “F5” tuşuna basarak veya yukarıda bulunan tuşa basarak programımızı başlatırız. Program başlatıldıktan sonra ise 



Şeklinde bir ekran gelecektir burada kamerayı elimle kapattığım için herhangi bir çıktı görememekteyiz eğer programımızı kapatmak ister isek “q” tuşuna basmalıyız yukarıdaki çarpı işareti ile kapatırız fakat program bir döngü içerisinde olduğundan kendini tekrar başlatacaktır.

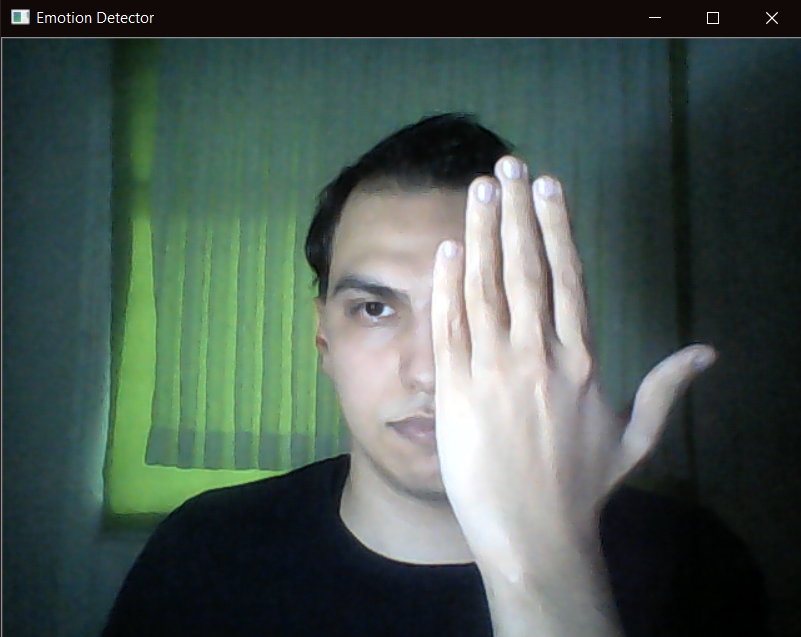
5

Kameramızı açtıktan sonra ise yüzümüz eğer net bir şekilde görünüyorsa programımız bu şekilde çalışacaktır.



Kullanıcının kafasını karıştırmamak için üzgün ifadesi dışındaki herhangi bir şeyi konsolda bastırmıyorum.

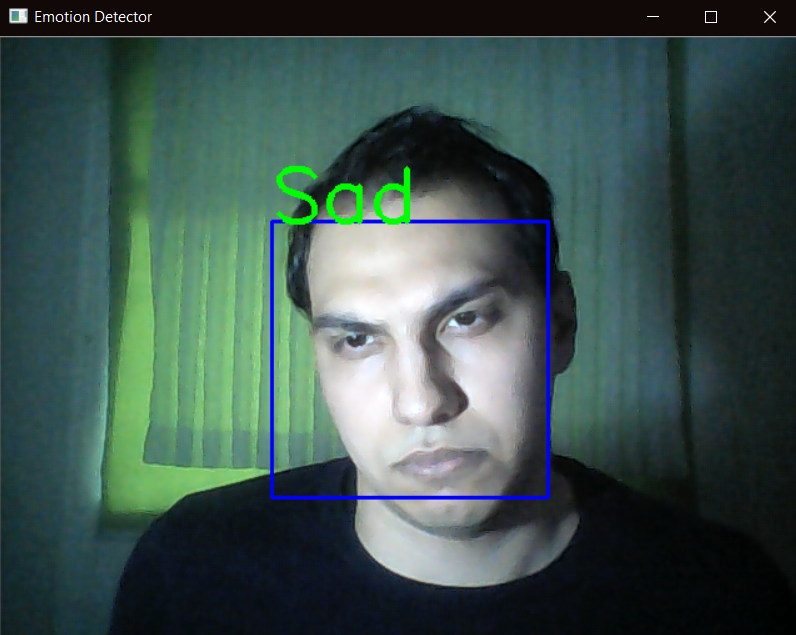
6



Eğer sistem yüzümüzü algılayamaz ise ekranda herhangi bir çıktı olmayacaktır

7

Eğer kullanıcı üzgün ise

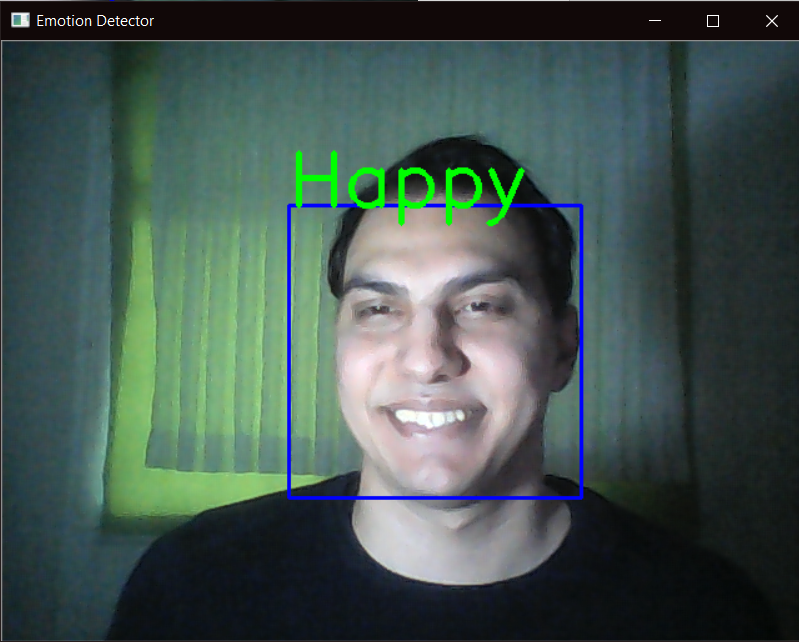


Konsola



Üzgün olduğu tarih ile birlikte saati bastırılır ve aynı zamanda bu “tarihler.txt” içerisine yazılır.

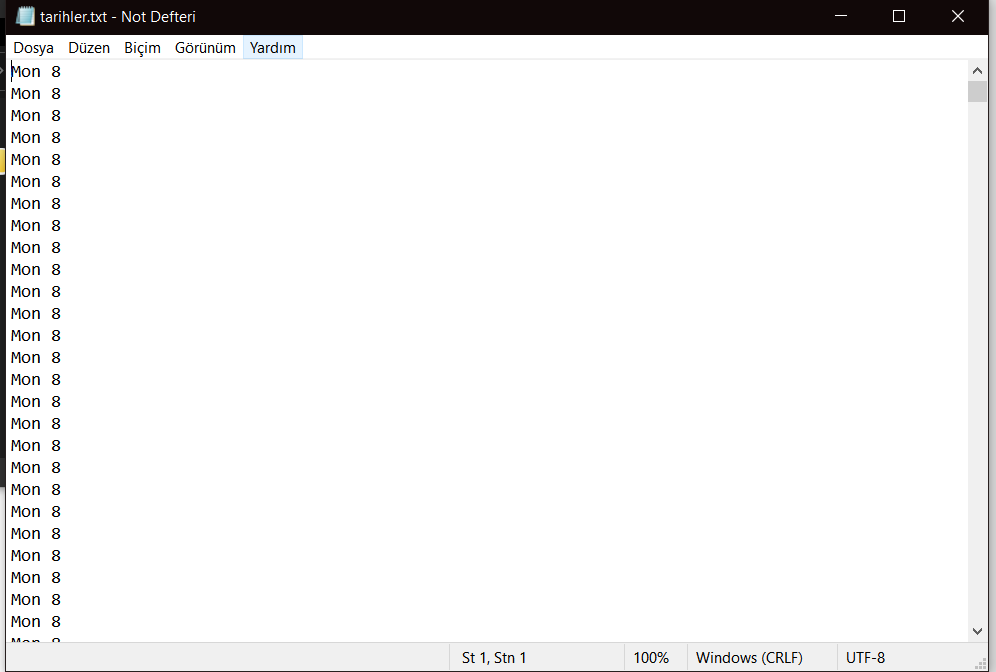
8



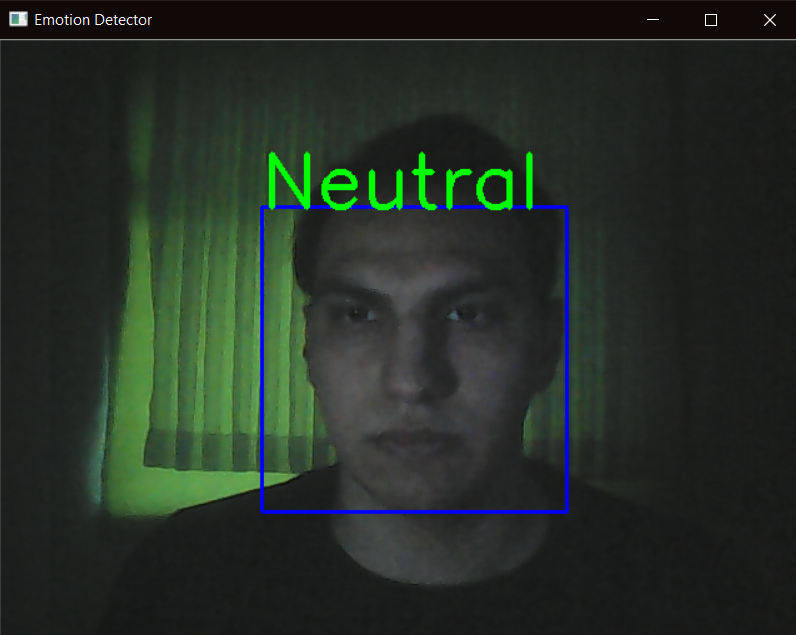
Gibi diğer ifadelerde konsola herhangi bir yazı bastırılmaz. eğer ki “tarih.txt” yoksa veya silinmişse önce oluşturur sonra eklenir.

9

“tarih.txt” dosyasının içi. Dosyanın içerisinde günleri kısaltarak hafızada daha az yer kaplamasını sağladım her bir karakterin 1 bayt olduğu ve alt satıra geçmek ise 2 bayt olduğunu göz önüne alarak böyle bir sistem yaptım. Peki neden hepsini aynı satıra yazdırıp virgüllerle ayırıp kod içerisinde de bu virgüllere göre ayırıp verileri de okuyabilirdik ama ben burada veri tabanının daha anlaşılır ve kodun daha anlaşılabilir olmasına odaklandım. Eğer sistem için kullanıcılar veriler çok yer kaplıyor der ise yukarıda bahsetmiş olduğum öneriyi sistem için kullanabiliriz hatta “mon 8“ yerine “mon8” şeklinde yazılarak da her gelen veri için 1 bayt tasarruf yapabiliriz.



10



Sistemimiz düşük ışık altında da çalışmaktadır fakat iyi bir ışıklandırma ile yüz mimiklerimiz daha rahat ayırt edilebilir.

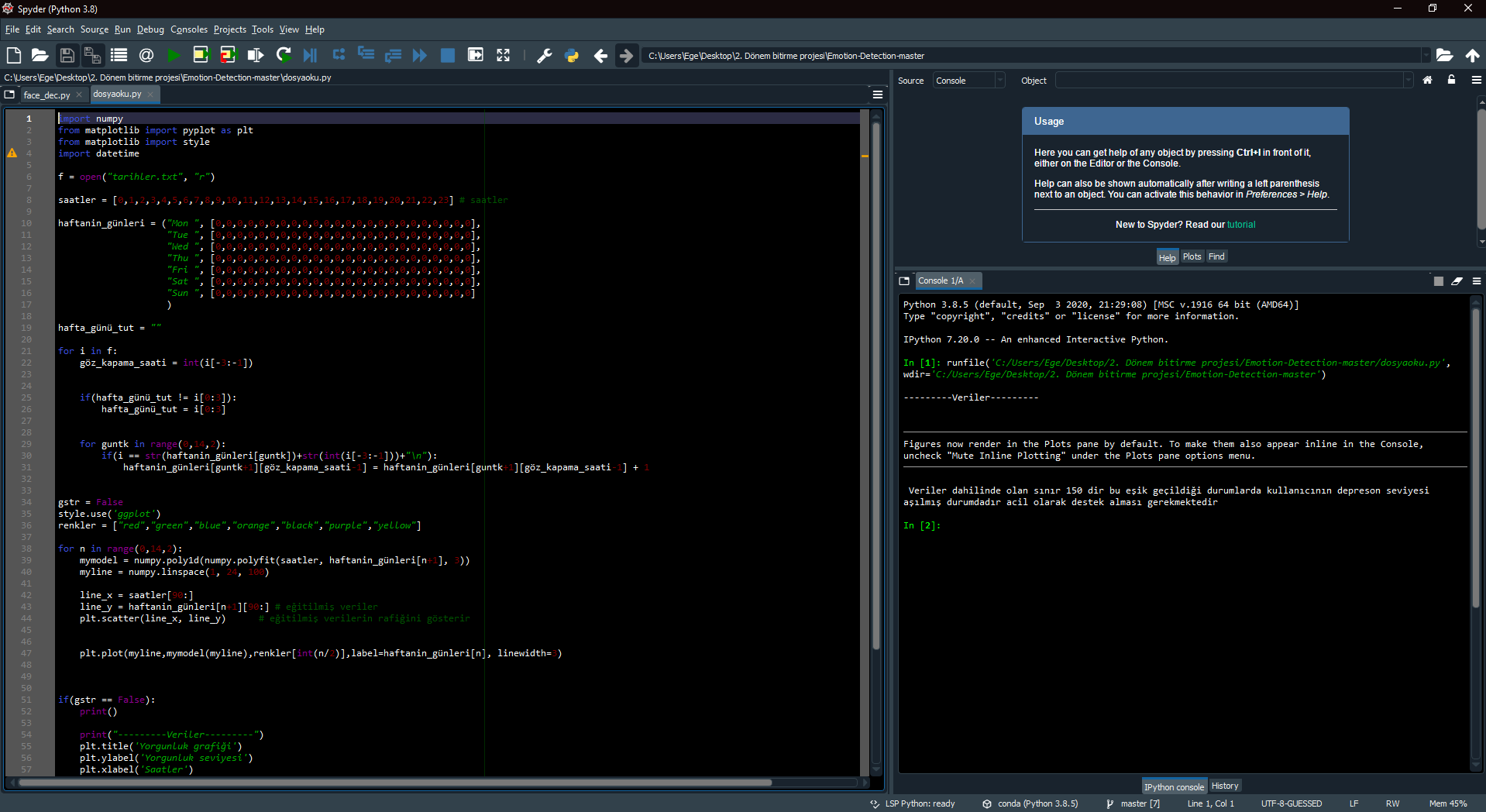


Kullanıcı yeterli ışıklandırmaya ve yeterli bir görüntü kalitesine sahip değil ise yüz algılanamaz.

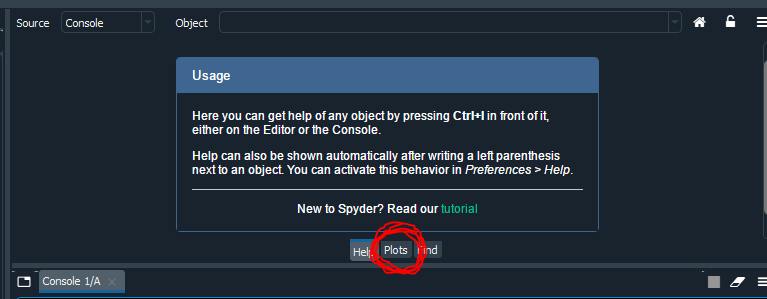
Kullanıcı “q” tuşuna bastıktan sonra program sonlanacaktır ve pencere kapanacaktır.

11

Sistem içerisine kaydedilen tüm bu verilerin analizini yapmak için ise “dosyaoku.py” klasörüne geçeriz ve çalıştırırız.

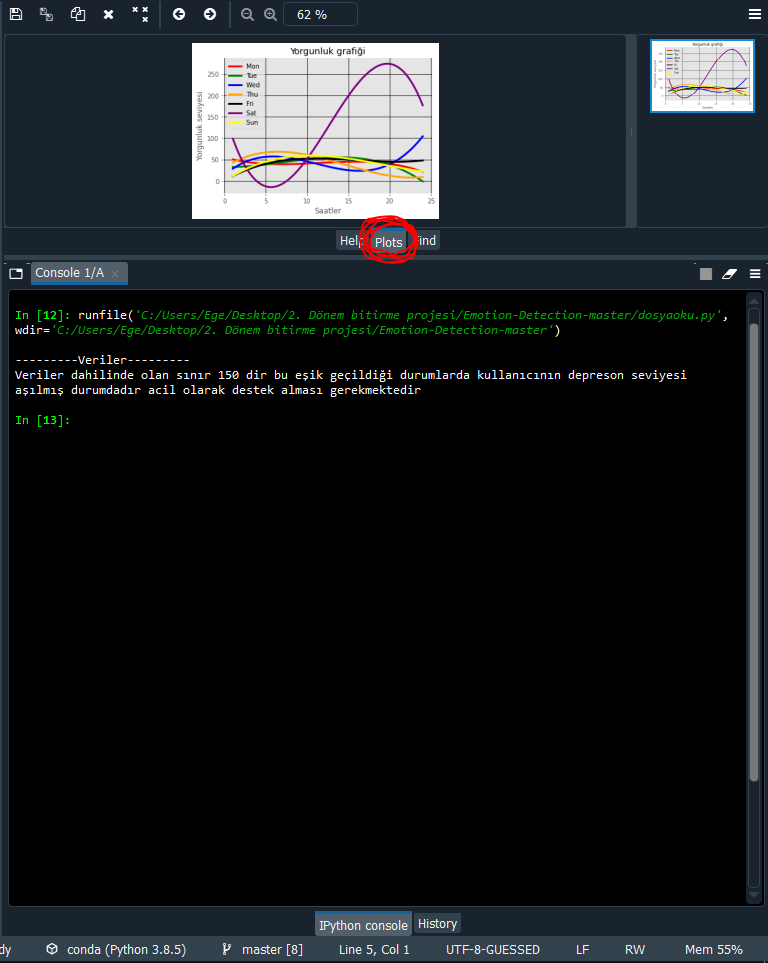


Çalıştırdıktan sonra ise “Plots” kısmını seçiyoruz.

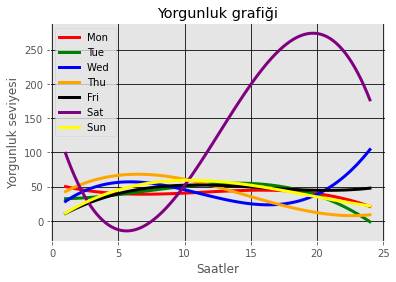


Ardından çıktımız sağ üstte görünecektir ve karşımıza aşağıdaki gibi bir çıktı gelecektir.

12

 (Plots kısmının yani kırmızı ile işaretlenmiş alanın seçili olduğundan emin olunuz yoksa grafiğimiz gözükmeyecektir.)

13



Aşağıda gördüğünüz grafik yukarıda bulunan grafiğin büyütülmüş halidir.

14

**Kaynak Kodlar**

**dosyaoku.py içerisindeki kodlar**

import pymongo

import numpy

from matplotlib import pyplot as plt

from matplotlib import style

import datetime

knt =input("Gerçek veriler için 1, test verileri için 2 yazınız : ")

f = open("tarihler.txt", "r")

hours = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23] # hours

week\_day = ("Mon ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Tue ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Wed ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Thu ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Fri ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Sat ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],

                    "Sun ", [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

                    )

keep\_day = ""

for i in f:

    sad\_face = int(i[-3:-1])

    if(keep\_day != i[0:3]):

        keep\_day = i[0:3]

    for guntk in range(0,14,2):

        if(i == str(week\_day[guntk])+str(int(i[-3:-1]))+"\n"):

            week\_day[guntk+1][sad\_face-1] = week\_day[guntk+1][sad\_face-1] + 1

gstr = False

style.use('ggplot')

renkler = ["red","green","blue","orange","black","purple","yellow"]

for n in range(0,14,2):

    mymodel = numpy.poly1d(numpy.polyfit(hours, week\_day[n+1], 3))

    myline = numpy.linspace(1, 24, 100)

    if(knt == "1"):

        train\_x = hours[:80]

        train\_y = week\_day[n+1][:80] #gerçek veriler

        plt.scatter(train\_x, train\_y)  # gerçek verilerin grafiğini gösterir

    elif(knt == "2"):

        test\_x = hours[80:]

        test\_y = week\_day[n+1][80:] # eğitilmiş veriler

        plt.scatter(test\_x, test\_y)      # eğitilmiş verilerin rafiğini gösterir

    else:

        print("Geçersiz değer")

        gstr = True

        break

    plt.plot(myline,mymodel(myline),renkler[int(n/2)],label=week\_day[n], linewidth=3)

if(gstr == False):

    print()

    if(knt == "1"):

        print("---------Gerçek Veriler---------")

        print(week\_day)

        print("------------------------")

    else:

        print("---------Test Verileri---------")

    plt.title('Yorgunluk grafiği')

    plt.ylabel('Yorgunluk seviyesi')

    plt.xlabel('hours')

    plt.legend()

    plt.grid(True,color='k')

    plt.show()

**face\_dec.py içerisindeki kodlar**

from keras.models import load\_model

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

from keras.preprocessing import image

import cv2

import numpy as np

import datetime

face\_classifier = cv2.CascadeClassifier('./haarcascade\_frontalface\_default.xml')

classifier =load\_model('./Emotion\_Detection.h5')

class\_labels = ['Angry','Happy','Neutral','Sad','Surprise']

cap = cv2.VideoCapture(0)

f = open("tarihler.txt", "a")

while True:

    ret, frame = cap.read()

    labels = []

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

    faces = face\_classifier.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:

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

        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)

            preds = classifier.predict(roi)[0]

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

            if(label == "Sad"):

                tarih = datetime.datetime.now().strftime("%A")

                saat = datetime.datetime.now().strftime("%H")

                f.write(str(tarih)[0:3]+ " " +str(saat) + "\n")

                print ("Tarih : " ,tarih);

                print ("Tarih Saat :" ,saat);

            label\_position = (x,y)

            cv2.putText(frame,label,label\_position,cv2.FONT\_HERSHEY\_SIMPLEX,2,(0,255,0),3)

        else:

            cv2.putText(frame,'No Face Found',(20,60),cv2.FONT\_HERSHEY\_SIMPLEX,2,(0,255,0),3)

    cv2.imshow('Emotion Detector',frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

f.close()

cap.release()

cv2.destroyAllWindows()