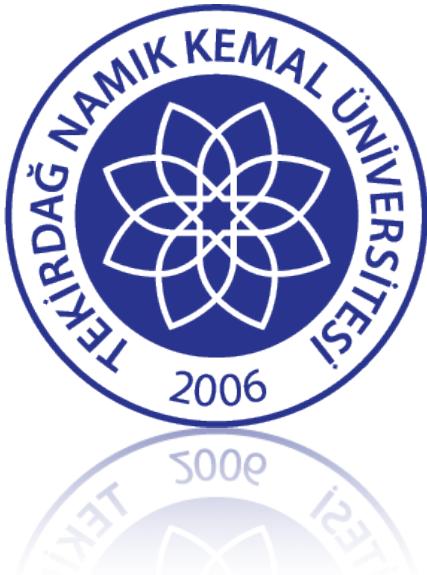


**TC NAMIK KEMAL ÜNİVERSİTESİ ÇORLU
MÜHENDİSLİK FAKÜLTESİ**



**BİLGİSAYAR MÜHENDİSLİĞİ PROJE-I DERSİ BİTİRME
ÇALIŞMASI**

YÜZ TANIMA İLE KAPI KİLİDİ SİSTEMİ

Ad ve Soyad : Yağmur Kahya

No : 2170656025

DANIŞMAN
Dr. Öğr. Üyesi Erkan ÖZHAN

ÖZET

Projenin amacı: Raspberry Pi 3 Model B+ kullanılarak yüz tanıma sistemli kapı kilidi tasarlanmıştır. Proje yüz tanıma teknolojisini kullanarak, erişimi kısıtlı, özel oda veya özel kasanın,sadece girişi izin verilen personel, kişinin girmesini amaçlar.

Kapıyı açmak isteyen kişi önceden sisteme tanıtılmış ise, sistem bu kişiyi tanırsa kapı kilidi açılır.Eğer kapıyı açmak isteyen kişi tanınmıyorsa, kapı kilidi açılmaz ve otomatik olarak ev sahibi kişiye, “Tanınmayan bir kişi kapıyı açmak istiyor “ yazan ve kapıdaki kişinin fotoğraflarını içeren bir e-mail gönderilir.

Projenin ilerleyışı: Projeyi tasarlamak için ilk olarak Raspberry Pi kartının kurulumunun yapılması gerekmektedir.Kurulumun nasıl yapıldığı detaylı olarak anlatılacaktır.Daha sonra Raspberry Pi kartını güce bağlamak ve bir kamera modülü bağlamak gerekmektedir. Kapı kilidi ve 5V röle kartı Raspberry Pi ‘nın GPIO pinlerine bağlanır.Sistemin çalışması için gerekli olan yazılım, Python dilinde yazılır ve gerekli kütüphaneler import edilir.E-mail alarm sistemi için de SMTP (**Simple Mail Transfer Protocol**) kullanılır.

Anahtar Kelimeler: Raspberry Pi, SMTP, OpenCV

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET.....	2
İÇİNDEKİLER.....	3
ŞEKİL LİSTESİ.....	5
ÖNSÖZ.....	6
1.GİRİŞ.....	7
1.1.Projenin Tanımı.....	7
1.2.Projenin Neden ve Amaçları.....	9
2.METERYAL VE YÖNTEM.....	10
2.1.Proje Gereksinimleri.....	10
2.1.1.Raspberry Pi.....	10
2.1.2.Raspberry Pi Kurulum.....	11
2.1.3.Selenoid Kilit.....	16
2.1.4.5V RÖLE KARTI VE 12 V GÜC KAYNAĞI.....	17
2.1.5.JUMPER KABLOLAR.....	17
2.2.1. YAZILIM GEREKSİNİMLERİ	19
2.2.2.MAKİNE ÖĞRENMESİ	19
2.2.3.PYTHON.....	19
2.2.4.OPENCV KÜTÜPHANESİ	20

2.2.5.NUMPY KÜTÜPHANESİ.....	20
2.2.6.HAAR CASCADE SINIFLANDIRICI.....	21
2.2.7.STMP(Simple Mail Transfer Protocol)	22
2.2.8.RASPBERRY Pİ GPIO İŞLEMLERİ	23
3.1.KODLAR.....	25
3.1.2.Yeni Kullanıcı Eklenmesi.....	26
3.1.3.Modeli Eğitme.....	27
3.1.4.Yüz Tanımalı Kapı Kilidi.....	28
3.1.5.GPIO Pinleri İle Selenoid Kilit Açma.....	30
3.1.6.SMTP Mail Gönderme.....	31
3.1.7.SMTP ile Mail Gönderirken Fotoğraf Çekme Fonksiyonu.....	32
SONUÇ VE ÖNERİLER.....	33
KAYNAKÇA.....	34

ŞEKİL LİSTESİ

Sayfa

Şekil1. Sistemin Genel Yapısı.....	9
Şekil2. Sistemin Çalışma Prensibi.....	9
Şekil4. Raspberry Pi ve SD Kart	12
Şekil5. USB kamera ve güç adaptörü bağlantıları	12
Şekil6. SD Kart Formatter.....	13
Şekil7. Raspberry Pi Imager	13
Şekil8. SSH ve wpa_supplicant.conf dosyaları.....	14
Şekil9. VNC Viewer	15
Şekil10. sudo raspi-config sistem ayarları sayfası	16
Şekil11. VNC viewer raspberry pi bağlantı	16
Şekil12. VNC viewer giriş ekranı	17
Şekil.13. Selenoid Kilit.....	17
Şekil14. 12V Akü ve 5V Röle Kartı.....	18
Şekil15. Jumper Kablolar.....	18
Şekil16. Röle,Akü ve Selenoid kilit bağlantısı.....	19
Şekil17. ML.....	20
Şekil18. OpenCV.....	21
Şekil19. Haar Cascade Classifiers	22
Şekil20. Google hesap güvenlik ayarı	23
Şekil21. GPIO pinlerinin GPIO numaraları ve fiziksel sıralaması...	24

ÖNSÖZ

Proje çalışmamın planlanmasında, yürütülmesinde ilgi ve desteğini esirgemeyen, Dr. Öğr. Üyesi Erkan ÖZHAN'a, Bilgisayar Mühendisliği Bölüm Öğretim Elemanlarına ve Tekirdağ Namık Kemal Üniversitesi'ne en içten teşekkürlerimi sunarım.

Ayrıca eğitimimiz süresince bize her konuda tam destek veren aileme ve bize bilgiler kazandıran tüm hocalarıma saygı ve sevgilerimi sunarım.

Tarih

10.01.2021

Ad Soyad

Yağmur Kahya

1.GİRİŞ

Yüz Tanıma Sistemi Kapı Kilidi projesinde Raspberry Pi 3 Model B+ kartı, Selenoid kilit, 5V röle kartı, USB Kamera, 12V akü kullanılmıştır.

Yüz Tanıma ile Kapı Kilidinin açılması hedeflenmiştir.Sistem yazılımı Python dilinde yazılmıştır.Model eğitimi sağlanmıştır.

1.1.Projenin Tanımı

Yüz Tanıma Sistemi Kapı Kilidi projesinde Raspberry Pi 3 Model B+ kartı kullanılmıştır. Selenoid kapı kilidini Raspberry Pi kartına bağlamak için 5V röle kartı ve 12V akü kullanılmıştır.

Çünkü; Raspberry Pi'nin GPIO pinleri 3,3V'luk bir çıkış verebilir ancak solenoid kilidin çalışması için 7-12V gereklidir. Bu nedenle, kilidi çalıştırmak için harici bir güç kaynağı ve röle kullanmamız gereklidir.

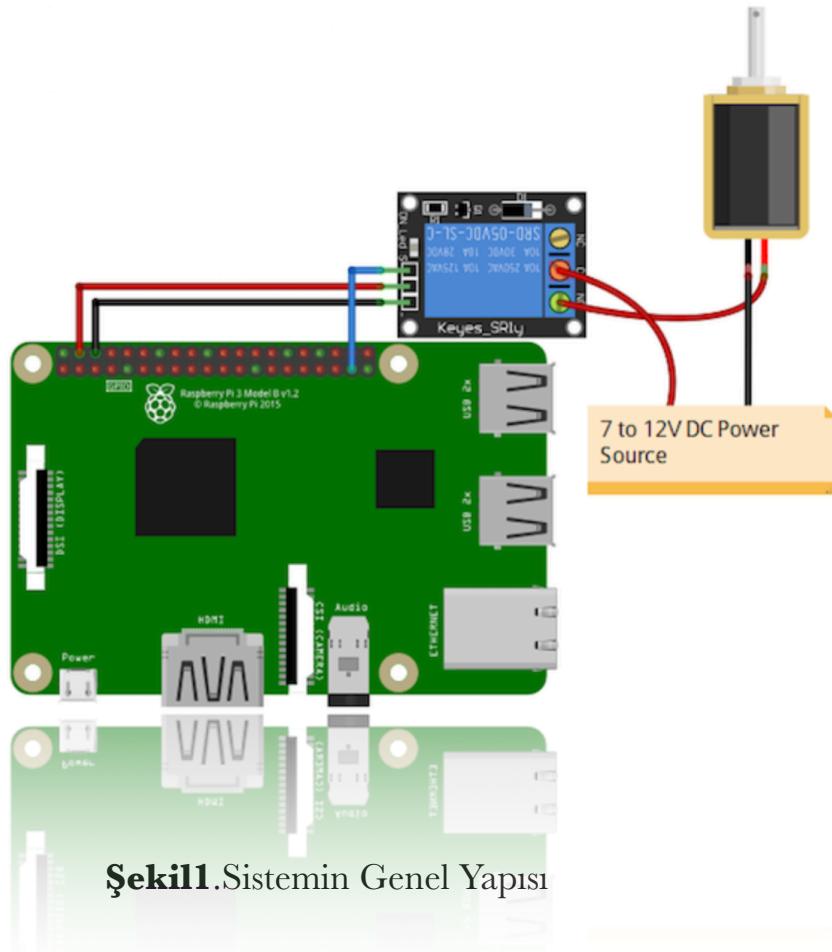
USB Kamera Raspberry Pi kartına bağlanır.

Ve Raspberry Pi kartı güce bağlanır.

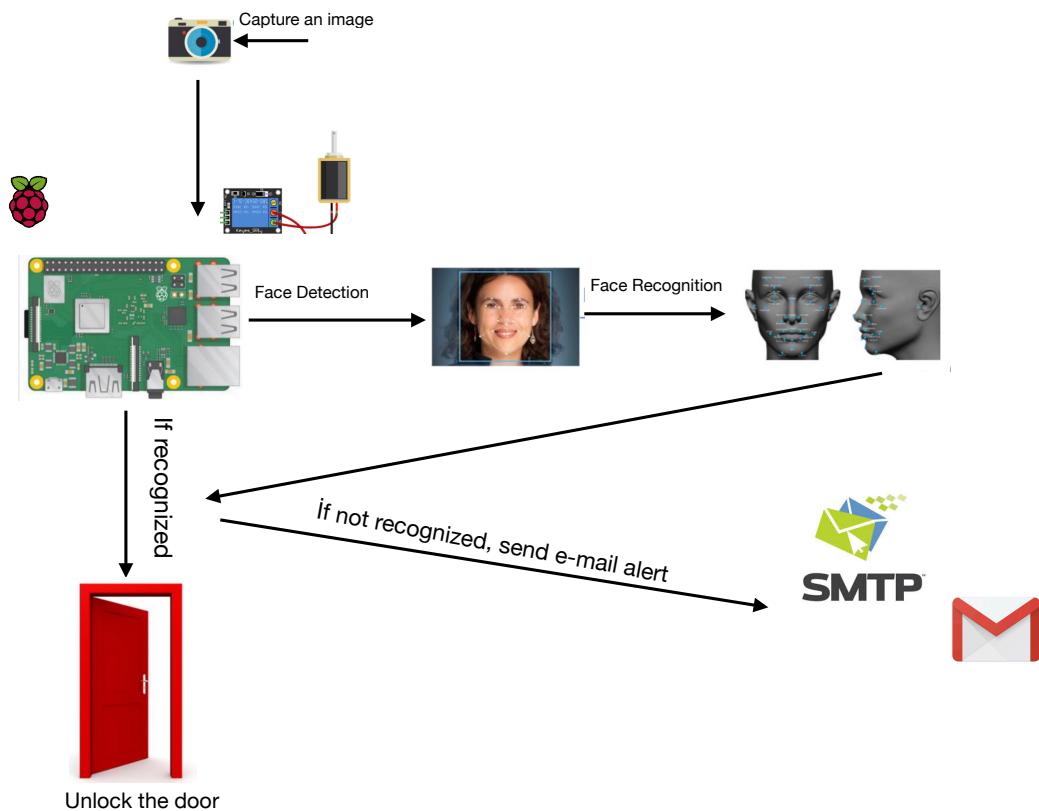
Kendi bilgisayarımız üzerinde gerekli kurulumlar yapıldıktan sonra VNC Viewer ile Raspberry Pi'a bağlanılır.

Raspbian işletim sistemi üzerine sistemin çalışması için gerekli olan Python kodları yazılmıştır.

1. Yazılımda Yüz Algılama ve Veri toplanması işlemleri yapılır.
2. Modelimizin eğitimi gerçekleştirilir. Eğitimden sonra model yüzleri algılayabilir hale gelecektir.
3. Model eğitiminden sonra Yüz Tanıma kodları yazılır ve model önceden öğrenmiş olduğu yüzleri tanır ve isimlerini yazdırır.
4. Modelin tanımadığı kişiler için e-mail göndermesini sağlayan kodlar yazılır.
5. Model tanıdığı kişiler için kilidi açar ancak tanımadığı kişilere kilidi açmaz ve kameraldan o kişinin görüntüsünü çekerek ev sahibi kişiye e-mail gönderir.



Şekil1.Sistemin Genel Yapısı



Şekil2.Sistemin Çalışma Prensibi

1.2.Projenin Nedeni ve Amaçları

Teknolojinin hızla gelişmesi ve internetin yaygınlaşması, ayrıca modern yaşamın ortaya çıkardığı ihtiyaçlar dolayısıyla Nesnelerin İnterneti (IoT)'ne olan ilgi gün geçtikçe artmaktadır.

Bu çalışmada akıllı güvenlik sistemlerinin bir parçası olarak algılanabilecek, raspberry pi ile yüz tanımlı kapı kilidi IoT sistemi geliştirilmiştir..

Son zamanlarda bilgisayarların, akıllı telefonların ve internetin kullanımı önemli ölçüde artmıştır. Bu artış insanların yaşamına kolaylıklar sağlamaşına rağmen birçok olumsuz yanı da hayatımıza girmiştir. Bu sorunlardan en önemlisini "güvenlik" oluşturmaktadır. Geliştirilen tüm akıllı sistemlerde bulunması gereken temel özellik, güvenliğin en üst düzeyde sağlanmasıdır.

Proje, erişimi kısıtlanmak istenen, sadece belirli kişilerin girişine izin verilmek istenen özel mekanlar (oda, kasa, çekmece) için tasarlanmıştır.

Eğer kildi açmak isteyen kişi tanınmayan biri ise, istenmeyen durumlara karşı, giriş yapmak isteyen bu kişinin bir fotoğrafı çekilir ve sistemde tanımlı adrese mail olarak gönderilir.

Özellikle, düşük maliyetli ve güvenlik ihtiyaçları gözetilerek geliştirilmiş, pratik olarak uygulanabilir bir çalışmadır.

2.METARYEL VE YÖNTEM

2.1.Proje Gereksinimleri

Projenin yapım aşamasında kullanılan metaryeller:

- 1.Raspberry Pi 3 Model B+
- 2.Selonoid Kilit
- 3.5V Röle Kartı
- 4.Jumpber Kablo
- 5.USB Kamera
- 6.Güç Adaptörü
- 7.12V Akü

2.1.1.Raspberry Pi

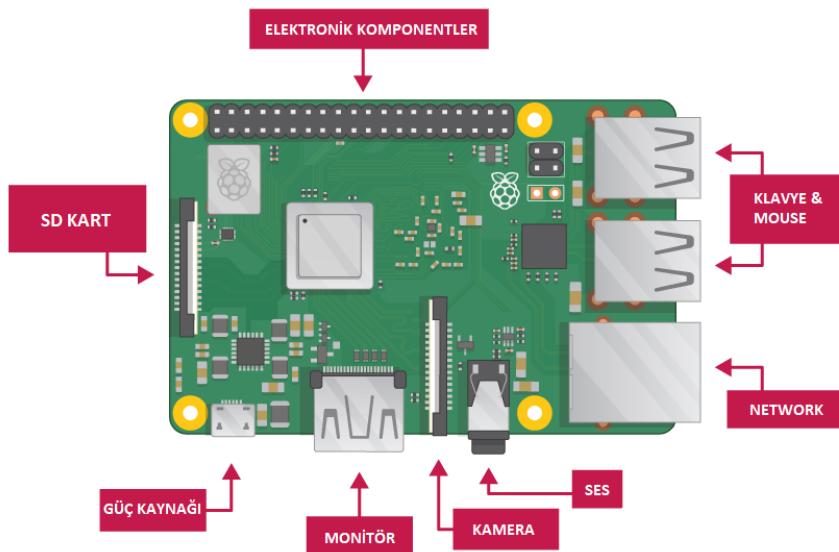
Raspberry Pi, okullarda ve gelişmekte olan ülkelerde temel bilgisayar bilimlerinin öğretilmesini teşvik etmek için geliştirilen küçük bir tek kartlı bilgisayar serisidir.

Herhangi bir monitöre takılır ve klavye, fare ve hoparlörler takılabilir.

Raspberry Pi, web'de gezinmek, belge ve elektronik tablo oluşturmak, oyun oynamak, video izlemek ve daha pek çok şey için kullanılabilir. Ayrıca, programlama ve dijital yapım öğrenimi için mükemmel bir ortam sağlar.

Raspberry Pi'nin GPIO (genel amaçlı giriş / çıkış) pinlerine çeşitli donanımlar bağlayabilir ve elektronik bileşenleri kullanarak programlamayı öğrenebilirsiniz.

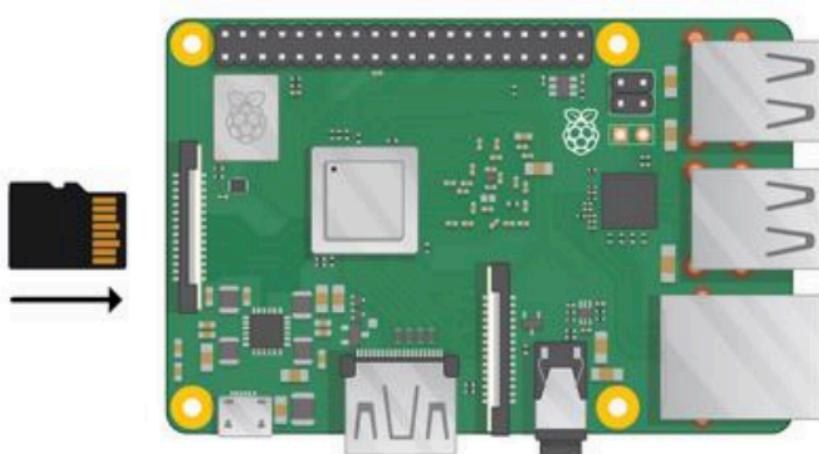
Raspberry Pi ayrıca IOT projeleri veya ev otomasyonu çözümleri gibi özel projelere de dahil edilebilir.



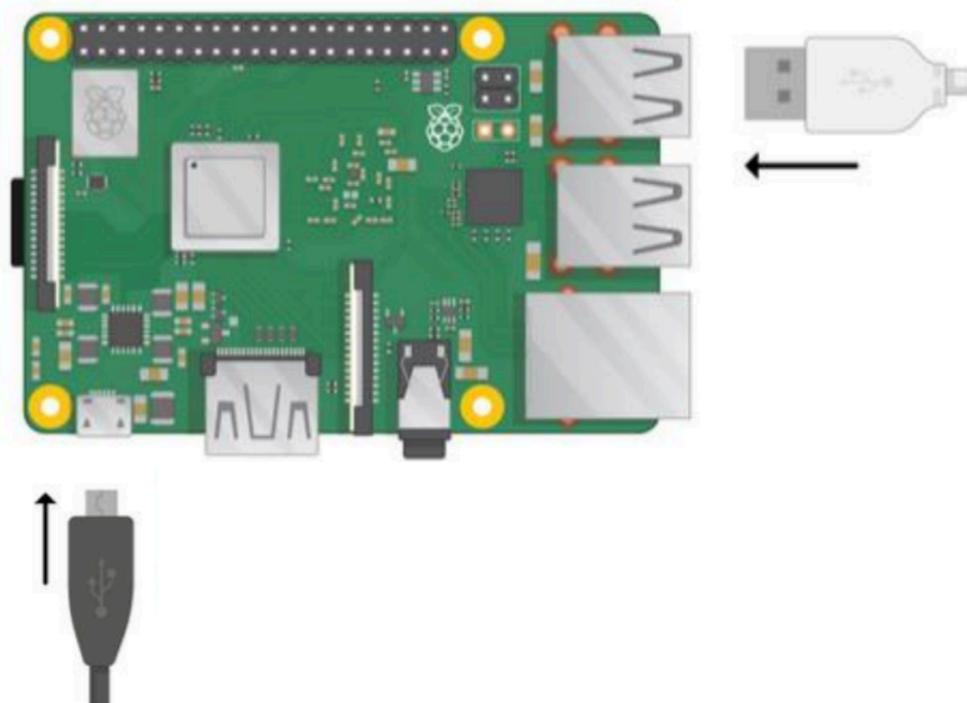
Şekil3.Raspberry Pi

2.1.2.Raspberry Pi Kurulum

Raspberry Pi üzerinde işlemler yapabilmek için öncelikle SD kart için gerekli işlemler tamamlanır ve Raspberry Pi'a takılır.



Şekil4.Raspberry Pi ve SD Kart



Şekil5.USB kamera ve güç adaptörü bağlantıları.

Raspberry Pi'a USB kamera bağlantısı şekil5'teki gibi yapılır ve kart güçe bağlanır.

Raspberry Pi mini bir bilgisayar olduğu için SD kart ile Raspberry Pi'ye özel geliştirilmiş **Raspberry Pi OS** işletim sistemi kurmamız gerekiyor. Bunun için;

1.SD Kart bilgisayara takılır ve SD kart formatter ile formatlanır.



Şekil6. SD Kart Formatter

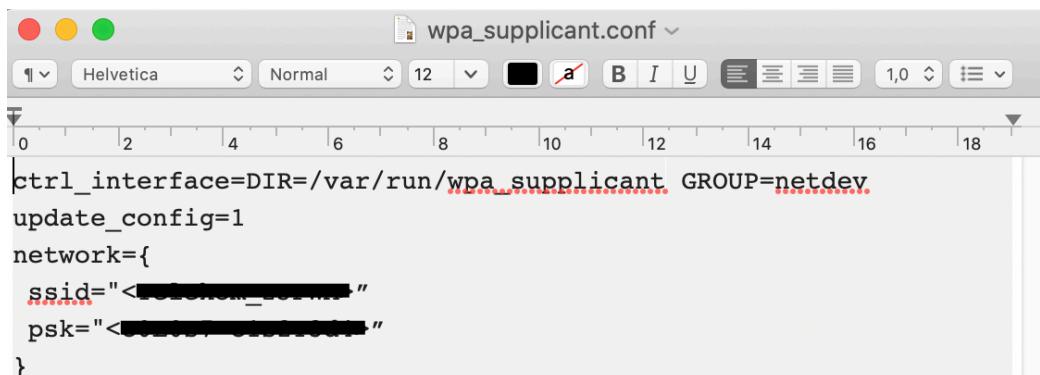
2.Raspberry Pi sitesinden Raspberry Pi Imager indirilir.



Programı çalıştırdıktan sonra gelen ekrandan sırayla **işletim sistemi** ve **kurulacağı diskı** seçiyoruz ve write tuşuna basıyoruz. **Write Successful** penceresi geldiğinde yazma işlemi **tamamlanmış** demektir. **Continue** tuşuna tıklayarak yazma işlemini bitiriyoruz. SD Kartı bilgisayardan çıkarıp tekrar takıyoruz.
Yazdırma işlemi tamamlandıktan sonra uzaktan bağlantı yapabilmek için SSH özelliğini aktif hale getirmemiz gerekiyor.

- Bunun için SD kartın içine boş bir metin belgesi oluşturup adını "ssh" olarak değiştiriyoruz.
- Dosya adı uzantısındaki ".txt" bölümünü silmemiz gerekiyor.
- Wifi bağlantısı için yine SD kartın içine bir metin belgesi oluşturuyoruz.
- Dosya adını "wpa_supplicant.conf" olarak değiştiriyoruz.
- Dosya adı uzantısındaki ".txt" bölümünü silmemiz gerekiyor.
- Oluşturduğumuz Metin belgesinin içine aşağıdaki bölüm yazıyoruz.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="MODEM ADINIZ"
    psk="MODEM ŞİFRENİZ"
}
```



Şekil8.SSH ve wpa_supplicant.conf dosyaları

3.RASPBERRY'E SSH İLE UZAKTAN BAĞLANMA

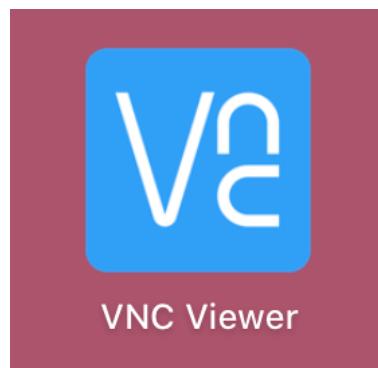
Mac/Linux için

- Bilgisayardan Terminal ekranı açılır
- ssh pi@192.168.?.? (raspberry'nin IP adresi) yazılıp Entera basılır
- Bağlanmak istiyor musunuz sorusu için "yes" yazılarak Entera basılır
- Şifre bölümüne raspberry'nin ilk kurulumdaki standart şifresi "raspberry" yazılıp Entera basılır
- Bağlantı gerçekleşikten sonra "sudo raspi-config" yazılarak Raspberry'nin sistem ayarları sayfası açılır

Windows için

- Bilgisayardan Terminal ekranı açılır (windowsta Putty uygulamasını kullanabilirsiniz)
- Login as bölümüne "pi" yazılıp Entera basılır
- Şifre bölümüne raspberry'nin ilk kurulumdaki standart şifresi "raspberry" yazılıp Entera basılır
- Bağlantı gerçekleşikten sonra "sudo raspi-config" yazılarak Raspberry'nin sistem ayarları sayfası açılır.

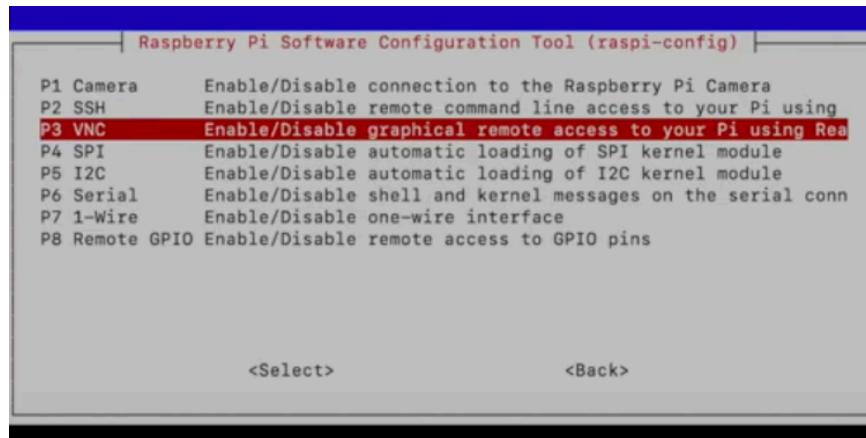
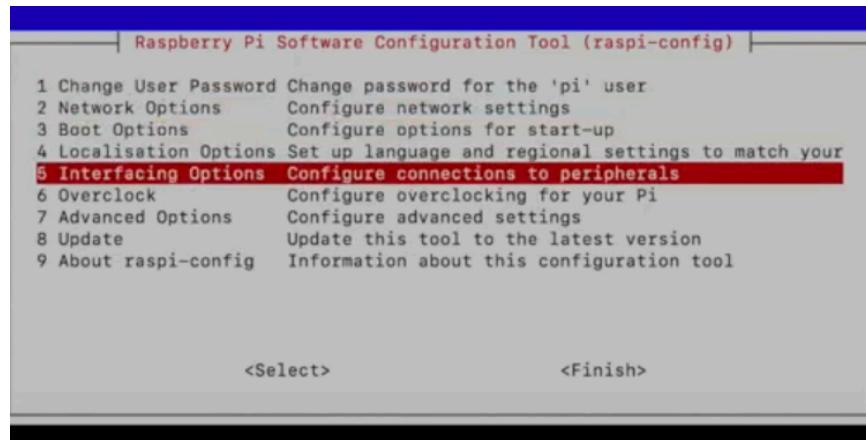
4.VNC Viewer Programı İndirilir



Şekil9.VNC Viewer

VNC VIEVER PROGRAMINI AKTİF HALE GETİRME

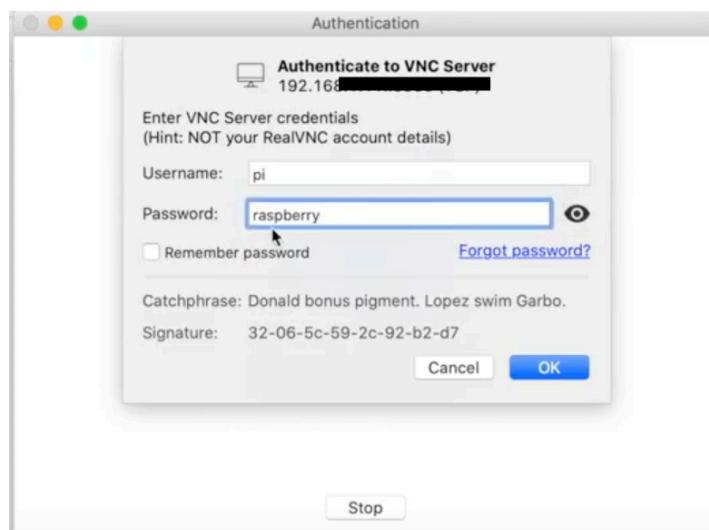
- Sistem ayarları sayfasından "Interfacing Options" bölümüne girilir
- VNC bölümüne girilir
- Açılan sayfanın alt bölümünden "YES" seçilerek kaydedilir.



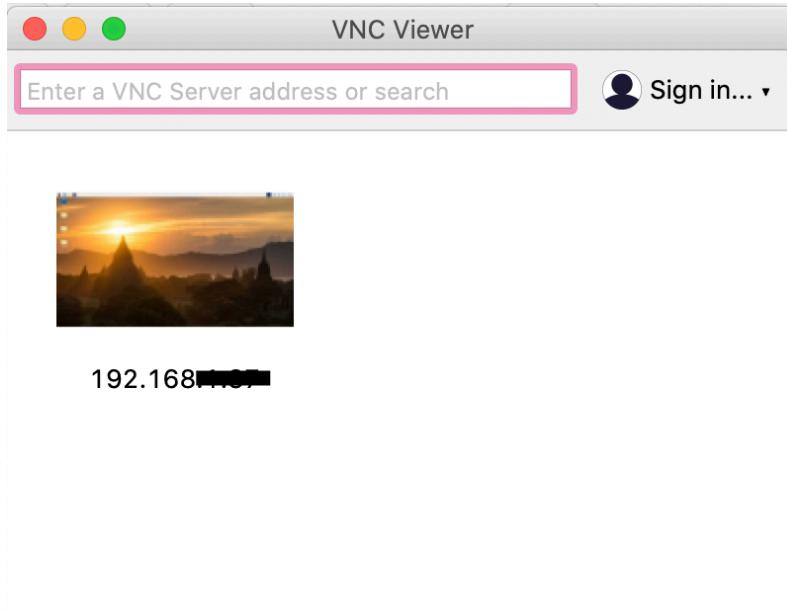
Şekil10.sudo raspi-config sistem ayarları sayfası

5.VNC UYGULAMASI İLE RASPBERRY MASAÜSTÜNE BAĞLANMA

- Uygulama açılır, sayfanın üst bölümündeki adres çubuğuuna Raspberry'nin bağlı olduğu IP adresi yazılır
- Açılan pencerede kullanıcı adına "pi" Şifreye "raspberry" yazarak Raspberry'e bağlanılır



Şekil11.VNC viewer raspberry pi bağlantı



Şekil12.VNC viewer giriş ekranı

2.1.3. SELENOİD KİLİT

Selenoid kilit, mandal bir voltaj uygulanarak otomatik olarak çalıştırılabilir. Solenoid kilidinde, bir kesinti (Basma düğmesi, Röle, vb.) etkinleştirildiğinde mandalı kapıya geri çeken düşük voltajlı bir selenoid bulunur. Mandal, kesme etkinleştirilene kadar konumunu koruyacaktır. Solenoid kilidin çalışma voltajı 12V'tur. 9V de kullanabilirsiniz, ancak bu daha yavaş çalışmaya neden olur. Solenoid kapı kilitleri, herhangi bir insan çabası gerektirmeden işlemleri otomatikleştirmek için genellikle uzak alanlarda kullanılır.



Şekil13.Solenoid Kilit

2.1.4. 5V RÖLE KARTI VE 12 V GÜC KAYNAĞI

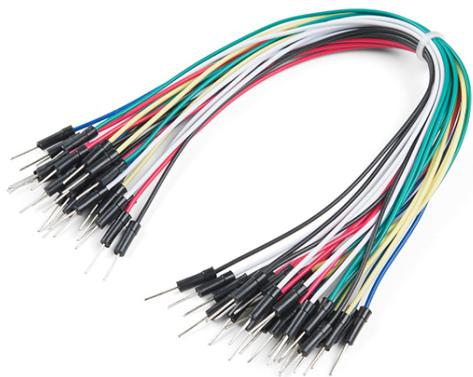
Solenoid Kilidin çalışması için 9V-12V gereklidir ve Raspberry Pi GPIO pinleri yalnızca 3.3V sağlayabilir, bu nedenle kilidi bir röle yardımıyla tetiklemek için 12V harici güç kaynağı kullanılır.



Şekil14.12V Akü ve 5V Röle Kartı

2.1.5.JUMPER KABLOLAR

Raspberry Pi ile Röle Kartını, selenoid kilidi ve güç kaynağını birbirine bağlamak için kullanılmıştır.



Şekil15.Jumper kablo

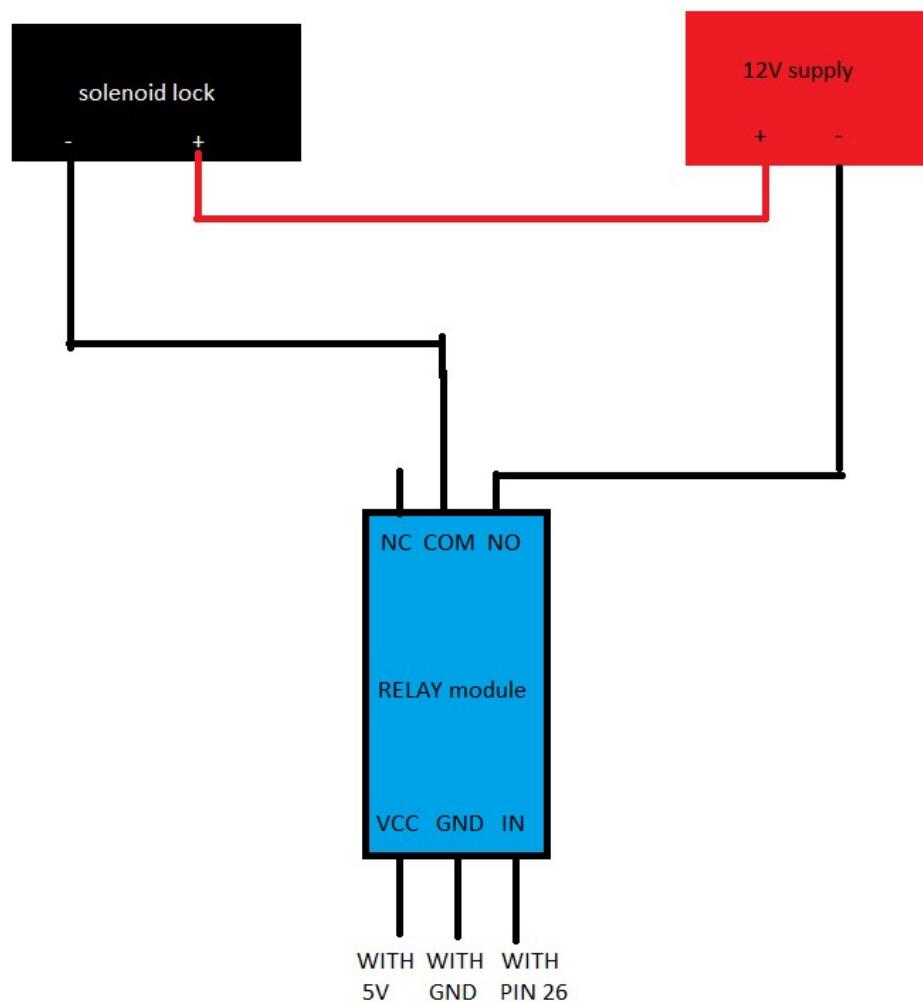
Röle modülünün bir tarafında üç, diğer tarafta üç olmak üzere toplam altı pini vardır. Alt tarafta sinyal, 5V ve toprak olmak üzere üç pin varıdır. Bu pinleri Raspberry Pi ile bağlayacağız. Diğer tarafta ise 5V rölesinin çıkış pinleri olan NC (Normalde kapalı), COM (Ortak) ve NO (normalde açık) vardır. Orada, 12V solenoid kildimizdeki çıkış cihazını bağlayacağız.

Raspberry Pi üzerindeki 5V pine VCC bağlanır.

Raspberry Pi üzerindeki Ground pine GND bağlanır.

IN pinini, Raspberry Pi GPIO26 pinine bağlanır.

**Aşağıdaki şekilde, solenoid kilit, 12 V akü ve röle kartının nasıl
bağlandığı gösterilmiştir.**

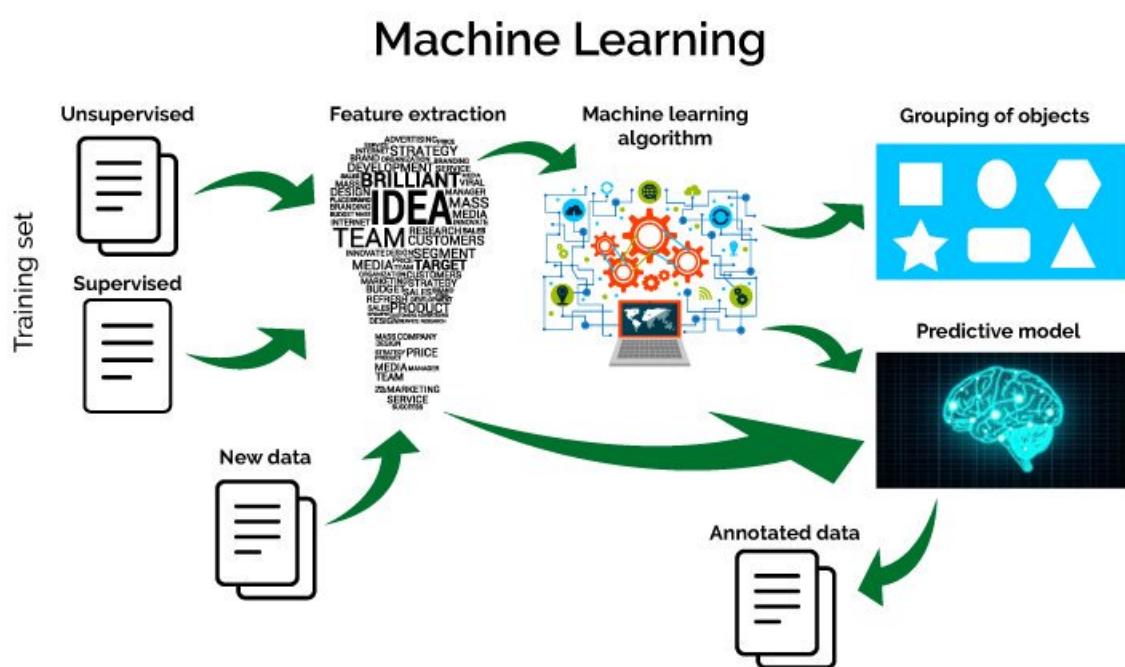


Şekil16.Röle,Akü ve Selenoid kilit bağlantısı

2.2.1. YAZILIM GEREKSİNİMLERİ

2.2.2. MAKİNE ÖĞRENMESİ

Makine Öğrenmesi, matematiksel ve istatistiksel işlemler ile veriler üzerinden çıkarımlar yaparak tahminlerde bulunan sistemlerin bilgisayarlar ile modellenmesidir.



Sekil17.ML

2.2.3. PYTHON

Python, ML için çok popüler bir dildir.

Veri temizleme, veriyi görselleştirme, derin öğrenme ve doğal dil işleme gibi veri analizleri için kullanılan çok zengin Python kütüphaneleri vardır.

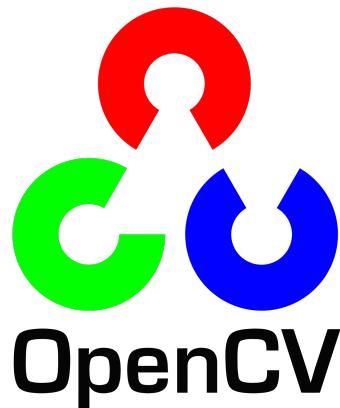
2.2.4. OPENCV KÜTÜPHANESİ

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir.

C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla kolaylıkla OpenCV uygulamaları geliştirilebilir.

OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500'den fazla algoritma bulunmaktadır.

Kolaylıkla matris işlemleri yapabileceğimiz Numpy kütüphanesi, OpenCV kütüphanesi yanında sıkılıkla tercih edilir.



Şekil18.OpenCV

2.2.5. NUMPY KÜTÜPHANESİ

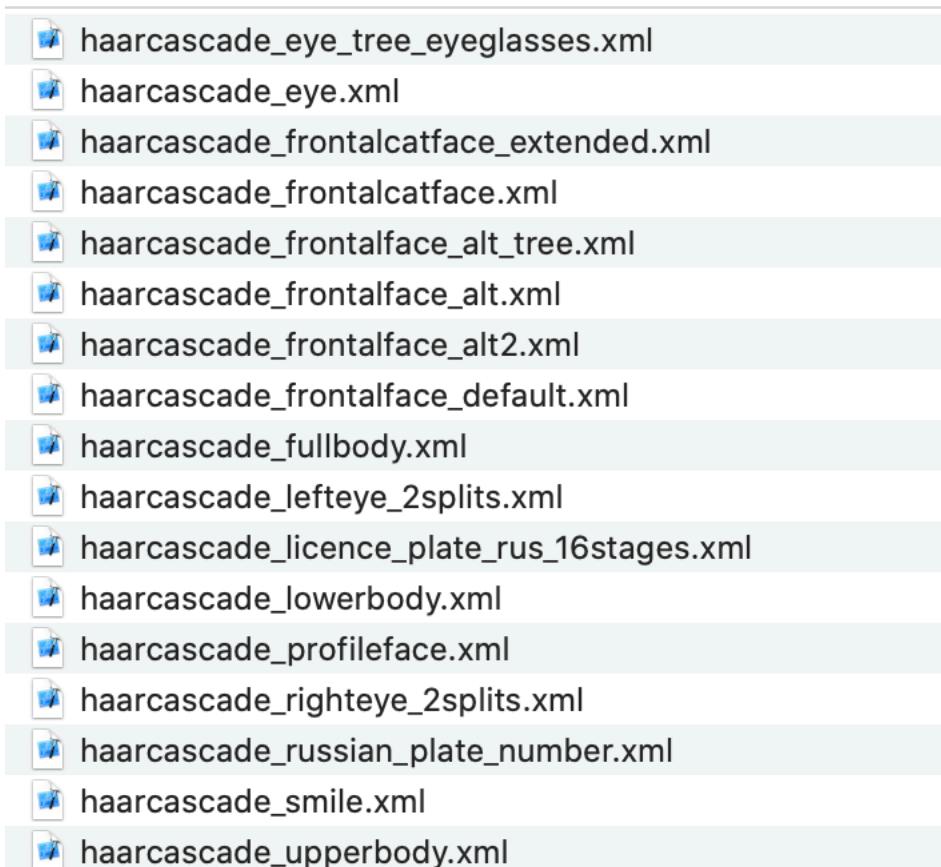
NumPy, Python' da bilimsel hesaplamlarda kullanılan temel pakettir. Çok boyutlu bir dizi nesnesi, çeşitli türetilmiş nesneler (maskelenmiş diziler ve matrisler gibi) ve dizilerdeki hızlı işlemler için matematiksel, mantıksal, şekil işleme, sıralama, seçme, G / Ç içeren bir dizi yordam sunan bir Python kitabıdır , Ayrık Fourier dönüşümü, temel doğrusal cebir, temel istatistik işlemler, rassal simülasyon ve çok daha fazlası.

2.2.6.HAAR CASCADE SINIFLANDIRICI

Haar Cascade, Paul Viola ve Michael Jones tarafından 2001 yılında "Boosted Cascade of Simple Features" başlıklı makalelerinde önerilen bir makine öğrenimi nesne algılama algoritmasıdır.

En temel manada belirli bir algoritmaya göre bulunması istenen nesneler önce bilgisayara tanıtılır ve model eğitilir. Daha sonra diğer görüntülerdeki nesneleri tespit etmek için kullanılır.

OpenCV, eğitim aldıkları görüntülere bağlı olarak kategoriler halinde (yüzler, gözler vb.) düzenlenmiş, önceden eğitilmiş Haar Cascade algoritmaları sunar.



Şekil19.Haar Cascade Classifiers

2.2.7. SMTP(Simple Mail Transfer Protocol)

SMTP (Basit Posta Aktarım Protokolü), sunucular arasında e-posta gönderip alırken kullanılan bir TCP / IP protokolüdür.

Sunucu yani Server, internet ortamında istekte bulunan bilgisayara yanıt vermek üzere özel olarak tasarlanmış bilgisayardır. İnternet ortamında veri paylaşımını sağlayan bu özel bilgisayarlardan **istekte bulunan bilgisayara ise İstemci denir**. Yüksek performansa sahip bu bilgisayarlar işlemleri kısa sürede gerçekleştirdiğinden tüm süreç saniyeler içerisinde gerçekleşir.

Python ile mail göndermek için smtplib ve email modüllerini kullanmamız gerekiyor. Mail sunucuları smtp protokolünü kullanarak email gönderip alırlar. Mail gönderebilmek için bir smtp mail sunucusuna bağlanmamız gerekmektedir. Göndermeden önce google hesabınıza daha az güvenli uygulamaların bağlanabilmesine izin vermelisiniz.

Google Hesap

← Daha az güvenli uygulama erişimi

Bazı uygulamalar ve cihazlar, hesabınızı saldırılara karşı savunmasız hale getiren daha az güvenli oturum açma teknolojileri kullanır. Bu uygulamalar için erişimi kapatabilirsiniz (bunu yapmanızı öneririz). Risklere rağmen bunları kullanmak istiyorsanız erişimi açabilirsiniz. Google, kullanılmadığında bu özelliği otomatik olarak KAPATIR. [Daha fazla bilgi](#)

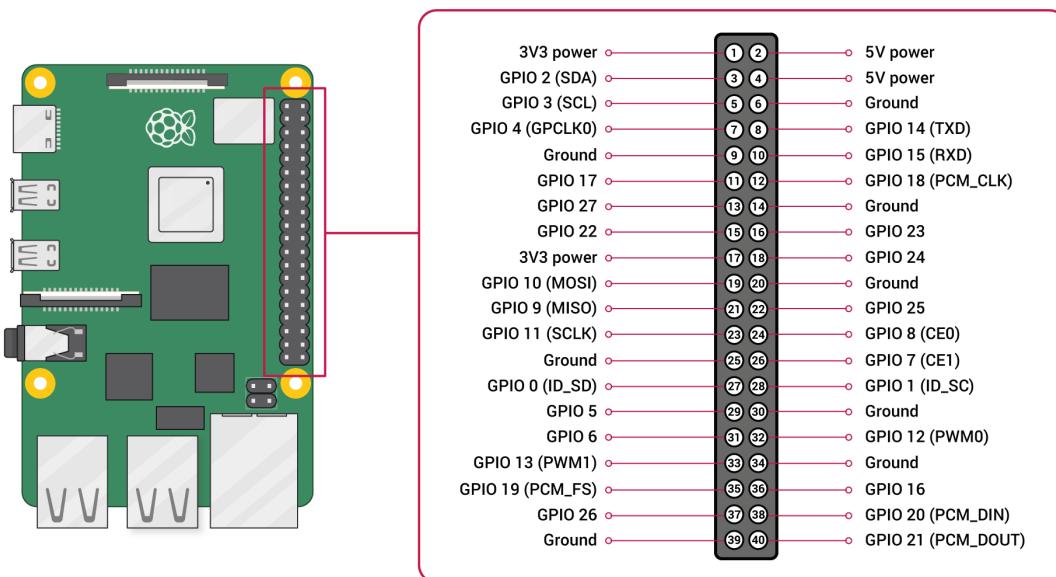
Daha az güvenli uygulamalara izin ver: AÇIK



Şekil20.Google hesap güvenlik ayarı

2.2.8.RASPBERRY PI GPIO İŞLEMLERİ

Raspberry Pi'ı yalnızca bir mini bilgisayar olarak değil, elektronik devrelerde kontrol kartı olarak da kullanabilmekteyiz. Bunu Raspberry Pi üzerinde bulunan GPIO pinlerini programlayarak sağlamaktayız. Raspberry Pi kartı üzerinde 40 tane GPIO pini bulunmaktadır. Bunlardan 2 tanesi 5V, 2 tanesi 3.3V, 8 tanesi GND ve 2 tanesi EEPROM için ayrılmış pinlerdir. Geriye kalan 26 pin Input/Output pinidir.



Şekil21.GPIO pinlerinin GPIO numaraları ve fiziksel sıralaması.

1. GPIO Kütüphanesinin Eklenmesi

```
import RPi.GPIO as GPIO
```

2. GPIO Pin Diziliminin Ayarlanması

Raspberry Pi'ın GPIO pinlerini isimlendirirken iki farklı dizilim vardır. Bunlar BCM dizilimi ve BOARD dizilimidir.

BCM dizilimi pinlere verilen GPIO numaralarından oluşmaktadır. Bunlar sıralı numaralar değildir.

BOARD dizilimi ise pinlerin fiziksel numaralandırılmasıdır. 1'den başlayıp 40'a kadar devam eden sıralı sayılarından oluşur.

GPIO pinlerini programlarken öncelikle GPIO pin dizilimini ayarlamalıyız. Bunun için aşağıdaki kod kullanılır:

`GPIO.setmode(GPIO.BCM)`

Veya

`GPIO.setmode(GPIO.BOARD)`

3. Bir Pinin Giriş veya Çıkış Olarak Ayarlanması

Dijital giriş/çıkış işlemlerinde öncelikle pinin giriş mi çıkış mı olacağını tanımlamalıyız. Bu tanımlama işlemini GPIO kütüphanesi içerisinde yer alan setup fonksiyonu ile gerçekleştirmektedir.

Bir pini giriş olarak ayarlamak için aşağıdaki kod kullanılır:

`GPIO.setup(pin_numarası,GPIO.IN)`

Bir pini çıkış olarak ayarlamak için aşağıdaki kod kullanılır:

`GPIO.setup(pin_numarası,GPIO.OUT)`

4. Bir Pinin Lojik-1 veya Lojik-0 Yapılması

Pinleri lojik-1 veya lojik-0 yapmak için `GPIO.output()` komutu kullanılmaktadır. Bu komutun kullanımı aşağıdaki gibidir:

`GPIO.output(pin_numarası,True)`

veya

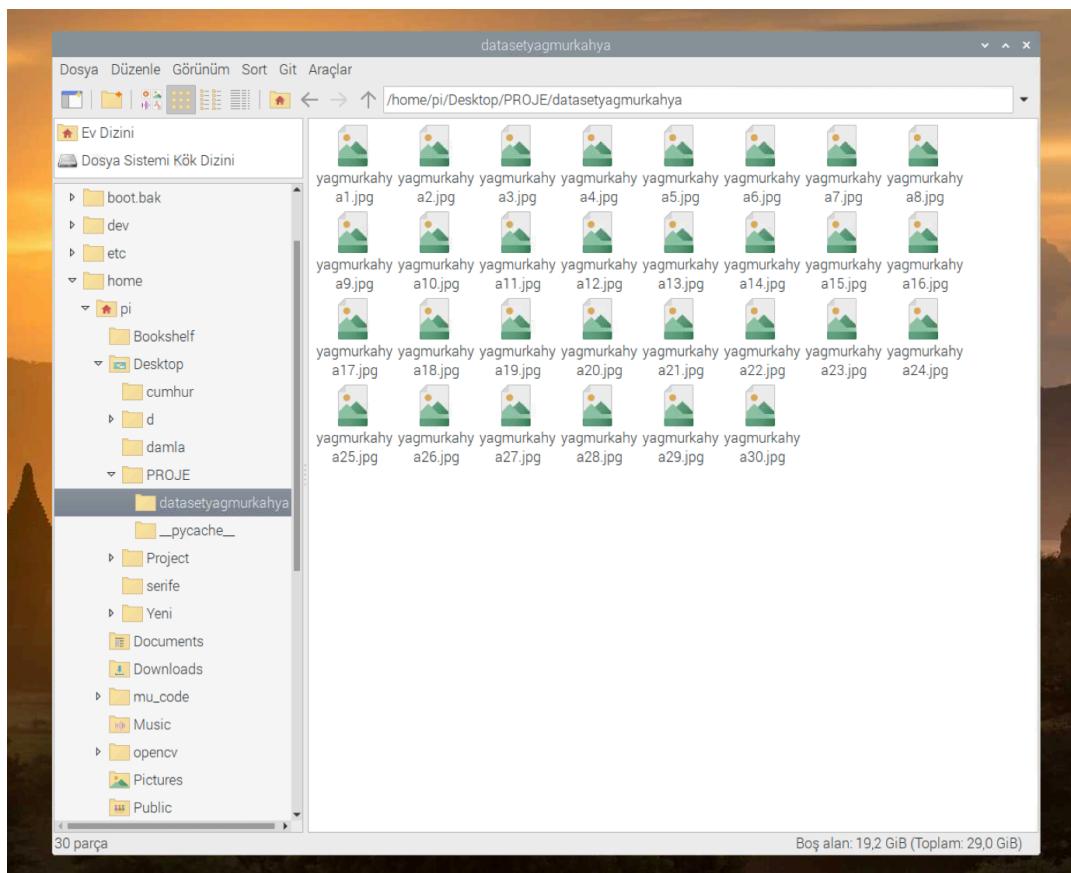
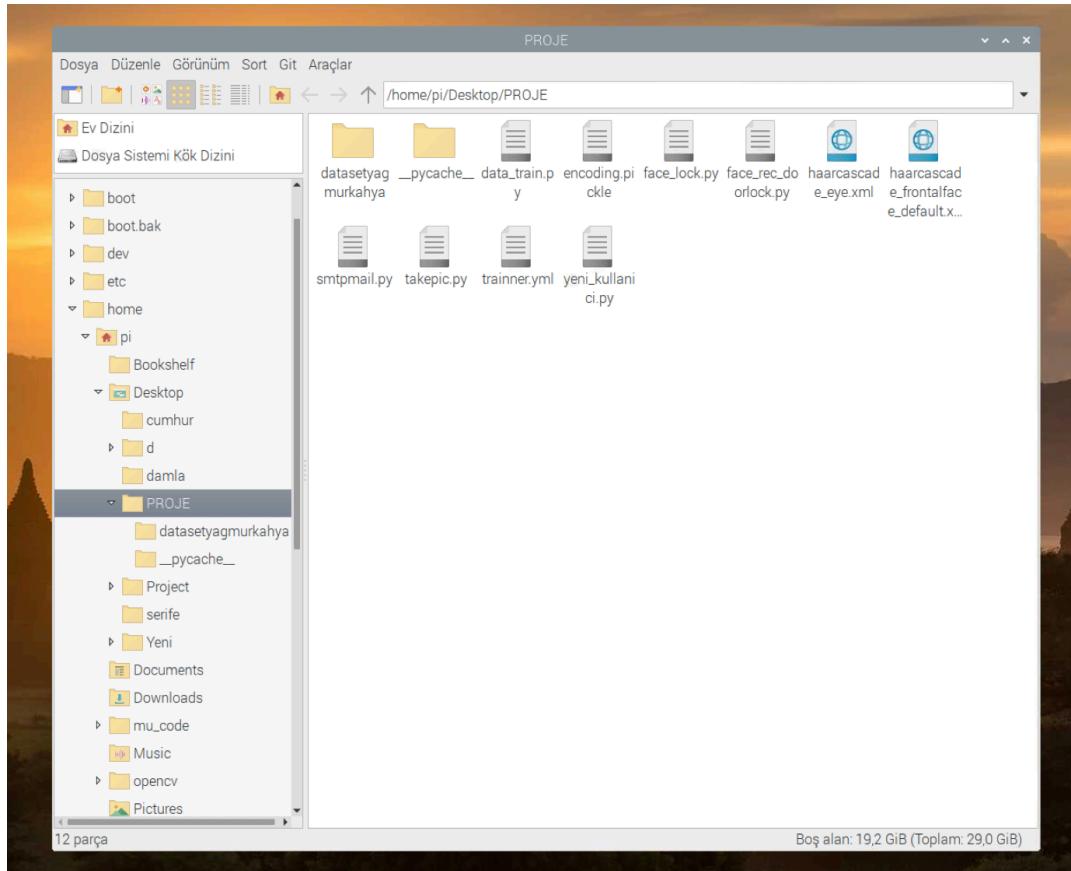
`GPIO.output(pin_numarası,False)`

5. Programa Gecikme Verilmesi

`import time`

`time.sleep(saniye)`

3.1.KODLAR



3.1.2. YENİ KULLANICI EKLEME

The screenshot shows the Mu 1.0.2 Python IDE interface. The title bar says "Mu 1.0.2 - yeni_kullanici.py". The toolbar has icons for Mode, New, Load, Save, Run, Debug, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar is a tab bar with several files: "yeni_kullanici.py", "data_train.py", "face_lock.py", "face_rec_doorlock.py", "smtpmail.py", and "takepic.py". The main code area contains the following Python script:

```
1 import cv2
2 import numpy as np
3 import os
4 import sys
5
6 path = os.path.dirname(os.path.abspath(__file__))
7 detector=cv2.CascadeClassifier(path+r'/home/pi/Desktop/y2tube/haarcascade_frontalface_default.xml')
8
9 camera = cv2.VideoCapture(0)
10 camera.set(3,640)
11 camera.set(4,480)
12
13 minW = 0.1*camera.get(3)
14 minH = 0.1*camera.get(4)
15
16 faceCascade = cv2.CascadeClassifier("/home/pi/Desktop/y2tube/haarcascade_frontalface_default.xml")
17
18 name = input("AD/SOYAD? ")
19 dirName = "/home/pi/Desktop/y2tube/dataset" + name
20 print(dirName)
21 if not os.path.exists(dirName):
22     os.makedirs(dirName)
23     print("Klasör oluşturuldu")
24 else:
25     print("İsim önceden kullanılmış")
26     sys.exit()
27
28 count = 1
29 while True:
30
31     if count > 30:
32         break
33     ret, im = camera.read()
34     gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
35     faces = faceCascade.detectMultiScale(gray, scaleFactor = 1.5, minNeighbors = 5)
36     for (x, y, w, h) in faces:
37         roiGray = gray[y:y+h, x:x+w]
38
39         fileName = dirName + "/" + name + str(count) + ".jpg"
40         cv2.imwrite(fileName, roiGray)
41         cv2.imshow("face", roiGray)
42         cv2.rectangle(im, (x, y), (x+w, y+h), (0, 255, 0), 2)
43
44         cv2.imshow('im', im)
45         key = cv2.waitKey(10)
46
47         if key == 27:
48             break
49
50
51 cv2.destroyAllWindows()
```

The screenshot shows the Mu 1.0.2 Python IDE interface, identical to the first one but with a few changes in the code. The title bar says "Mu 1.0.2 - yeni_kullanici.py". The toolbar and tabs are the same. The main code area now includes saving the captured images to a file and destroying all windows at the end of the loop:

```
16 faceCascade = cv2.CascadeClassifier("/home/pi/Desktop/y2tube/haarcascade_frontalface_default.xml")
17
18 name = input("AD/SOYAD? ")
19 dirName = "/home/pi/Desktop/y2tube/dataset" + name
20 print(dirName)
21 if not os.path.exists(dirName):
22     os.makedirs(dirName)
23     print("Klasör oluşturuldu")
24 else:
25     print("İsim önceden kullanılmış")
26     sys.exit()
27
28 count = 1
29 while True:
30
31     if count > 30:
32         break
33     ret, im = camera.read()
34     gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
35     faces = faceCascade.detectMultiScale(gray, scaleFactor = 1.5, minNeighbors = 5)
36     for (x, y, w, h) in faces:
37         roiGray = gray[y:y+h, x:x+w]
38
39         fileName = dirName + "/" + name + str(count) + ".jpg"
40         cv2.imwrite(fileName, roiGray)
41         cv2.imshow("face", roiGray)
42         cv2.rectangle(im, (x, y), (x+w, y+h), (0, 255, 0), 2)
43
44         cv2.imshow('im', im)
45         key = cv2.waitKey(10)
46
47         if key == 27:
48             break
49
50
51 cv2.destroyAllWindows()
```

3.1.3. MODELİ EĞİTME

```

Mu 1.0.2 - data_train.py

Mode New Load Save Run Debug REPL Plotter Zoom-in Zoom-out Theme Check Help Quit
yeni_kullanici.py x data_train.py x face_lock.py x face_rec_doorlock.py x smtpmail.py x takepic.py x

1 import os
2 import numpy as np
3 from PIL import Image
4 import cv2
5 import pickle
6
7 faceCascade = cv2.CascadeClassifier("/home/pi/Desktop/y2tube/haarcascade_frontalface_default.xml")
8 # OpenCV paketinde bulunan LBPH (YEREL İKİLİ PATTERN HISTOGRAMS) yüz tanıycı kullanıyoruz.
9 recognizer = cv2.face.LBPHFaceRecognizer_create()
10
11 # Geçerli çalışma dizininin yolunu alır ve images dizinine gidilir.
12
13 baseDir = os.path.dirname(os.path.abspath(__file__))
14 imageDir = os.path.join(baseDir, "datasetyagmurkaha")
15
16 currentId = 1
17 label_ids={}
18 y_labels=[]
19 x_train=[]
20
21 #Her görüntü dizinine gider ve görüntülerleri arar.
22 #Görüntü mevcutsa, NumPy dizisine dönüştürüyoruz
23 for root, dirs, files in os.walk(imageDir):
24     print(root, dirs, files)
25     for file in files:
26         print(file)
27         if file.endswith("png") or file.endswith("jpg"):
28             path = os.path.join(root, file)
29             label = os.path.basename(root)
30             print(label)
31
32             if not label in label_ids:
33                 label_ids[label] = currentId
34                 print(label_ids)
35             currentId += 1
36
37 #Doğru görüntülere sahip olduğumuzdan emin olmak için yüz algılamayı tekrar gerçekleştiriyoruz.

```

```

Mu 1.0.2 - data_train.py

Mode New Load Save Run Debug REPL Plotter Zoom-in Zoom-out Theme Check Help Quit
yeni_kullanici.py x data_train.py x face_lock.py x face_rec_doorlock.py x smtpmail.py x takepic.py x

21 #Her görüntü dizinine gider ve görüntülerleri arar.
22 #Görüntü mevcutsa, NumPy dizisine dönüştürüyoruz
23 for root, dirs, files in os.walk(imageDir):
24     print(root, dirs, files)
25     for file in files:
26         print(file)
27         if file.endswith("png") or file.endswith("jpg"):
28             path = os.path.join(root, file)
29             label = os.path.basename(root)
30             print(label)
31
32             if not label in label_ids:
33                 label_ids[label] = currentId
34                 print(label_ids)
35             currentId += 1
36
37 #Doğru görüntülere sahip olduğumuzdan emin olmak için yüz algılamayı tekrar gerçekleştiriyoruz.
38 #Ve sonra kıyaslama verilerini hazırlıyoruz
39         id_ = label_ids[label]
40         pilImage = Image.open(path).convert("L")
41         imageArray = np.array(pilImage, "uint8")
42         faces = faceCascade.detectMultiScale(imageArray, scaleFactor=1.1, minNeighbors=5)
43
44         for (x, y, w, h) in faces:
45             roi = imageArray[y:y+h, x:x+w]
46             x_train.append(roi)
47             y_labels.append(id_)
48 # Dizin adlarını ve etiket kimliklerini içeren sözlüğü saklayoruz.
49 with open("labels", "wb") as f:
50     pickle.dump(label_ids, f)
51     f.close()
52 # Verileri eğiterek ve dosyayı kaydediyoruz.
53
54 recognizer.train(x_train,np.array(y_labels))
55 recognizer.save("/home/pi/Desktop/y2tube/trainner.yml")
56 print(label_ids)

```

3.1.4. YÜZ TANIMA KAPI KİLİDİ

Mu 1.0.2 - face_rec_doorlock.py

```

1 import cv2,os
2 import numpy as np
3 import pickle
4 import RPi.GPIO as GPIO
5 import time
6 from PIL import Image
7 from face_lock import unlock,lock
8 import smtpçalişan as m
9
10 id = 0
11
12 names = ['None', 'yagmurkahya']
13
14
15 with open('labels', 'rb') as f:
16     dicti = pickle.load(f)
17     f.close()
18
19 camera = cv2.VideoCapture(0)
20 camera.set(3,640)
21 camera.set(4,480)
22 minW = 0.1*camera.get(3)
23 minH = 0.1*camera.get(4)
24
25 path = os.path.dirname(os.path.abspath(__file__))
26
27 faceCascade = cv2.CascadeClassifier("/home/pi/Desktop/y2tube/haarcascade_frontalface_default.xml")
28 recognizer = cv2.face.LBPHFaceRecognizer_create()
29 recognizer.read("/home/pi/Desktop/y2tube/trainer.yml")
30
31 font = cv2.FONT_HERSHEY_SIMPLEX
32 prevTime = 0
33 doorUnlock = False
34
35 while True:
36     ret, im =camera.read()

```

Mu 1.0.2 - face_rec_doorlock.py

```

37     ret, im =camera.read()
38     gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
39     faces = faceCascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5, minSize=(100, 100),flags=cv2.CASCADE_SCALE_IMAGE)
40     for (x, y, w, h) in faces:
41
42         cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 2)
43
44         id, confidence = recognizer.predict(gray[y:y + h, x:x + w])
45
46         # Check if confidence is less them 100 ==> "0" is perfect match
47         if (confidence < 100):
48
49             id = names[id]
50             confidence = " {0}%".format(round(100 - confidence))
51
52             print("Opening Lock")
53             unlock(26)
54             time.sleep(10)
55             lock(26)
56             GPIO.cleanup(26)
57
58
59         else:
60
61
62             id = "unknown"
63             confidence = " {0}%".format(round(100 - confidence))
64             lock(26)
65             m.mails()
66
67             cv2.putText(im, str(id), (x + 5, y - 5), font, 1, (255, 255, 255), 2)
68             cv2.putText(im, str(confidence), (x + 5, y + h - 5), font, 1, (255, 255, 0), 1)
69
70             cv2.imshow('camera', im)
71
72             k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
73             if k == 27:

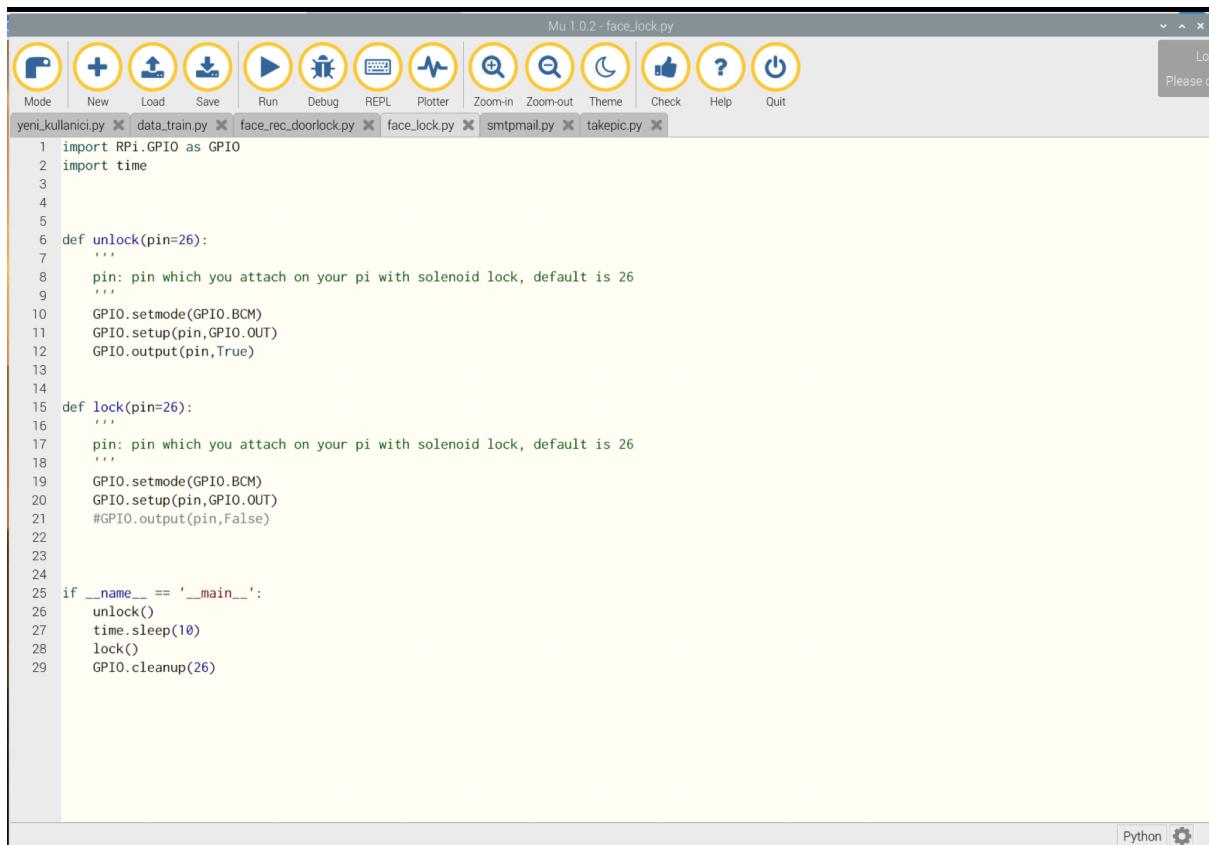
```

The screenshot shows the Mu 1.0.2 Python IDE interface. The title bar reads "Mu 1.0.2 - face_rec_doorlock.py". The menu bar includes "File", "Edit", "Run", "Tools", "Help", and "About". The toolbar contains icons for Mode, New, Load, Save, Run, Debug, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. The status bar on the right says "Lo Please c". The code editor displays the following Python script:

```
44     id, confidence = recognizer.predict(gray[y:y + h, x:x + w])
45
46     # Check if confidence is less than 100 ==> "0" is perfect match
47     if (confidence < 100):
48
49         id = names[id]
50         confidence = " {0}%".format(round(100 - confidence))
51
52         print("Opening Lock")
53         unlock(26)
54         time.sleep(10)
55         lock(26)
56         GPIO.cleanup(26)
57
58     else:
59
60
61         id = "unknown"
62         confidence = " {0}%".format(round(100 - confidence))
63         lock(26)
64         m.mails()
65
66         cv2.putText(im, str(id), (x + 5, y - 5), font, 1, (255, 255, 255), 2)
67         cv2.putText(im, str(confidence), (x + 5, y + h - 5), font, 1, (255, 255, 0), 1)
68
69         cv2.imshow('camera', im)
70
71         k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
72         if k == 27:
73             break
74
75
76     # Do a bit of cleanup
77     print("\n [INFO] Exiting Program and cleanup stuff")
78     camera.release()
79     cv2.destroyAllWindows()
```

The status bar at the bottom right shows "Python" and a gear icon.

3.1.5.GPIO PINLERİ SELENOİD KİLİT AÇMA-KAPAMA

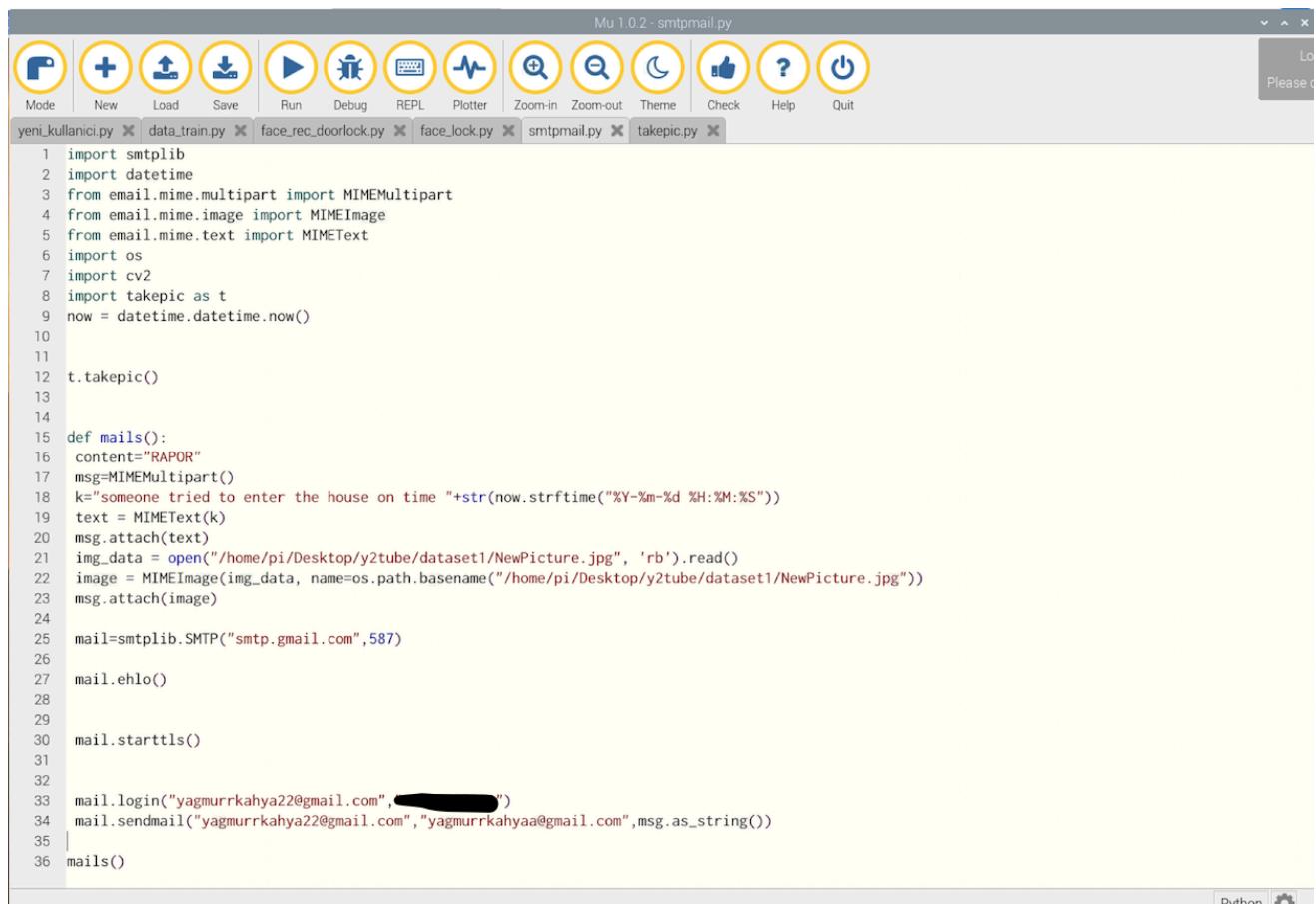


The screenshot shows the Mu 1.0.2 Python IDE interface. The title bar reads "Mu 1.0.2 - face_lock.py". The menu bar includes "File", "Edit", "Run", "Tools", "Help", and "About". The toolbar contains icons for Mode (New, Load, Save), Run (Run, Debug, REPL, Plotter), Zoom (Zoom-in, Zoom-out), Theme, Check, Help, and Quit. The status bar on the right says "Please connect a serial port". The code editor window displays the following Python script:

```
1 import RPi.GPIO as GPIO
2 import time
3
4
5
6 def unlock(pin=26):
7     """
8     pin: pin which you attach on your pi with solenoid lock, default is 26
9     """
10    GPIO.setmode(GPIO.BCM)
11    GPIO.setup(pin,GPIO.OUT)
12    GPIO.output(pin,True)
13
14
15 def lock(pin=26):
16     """
17     pin: pin which you attach on your pi with solenoid lock, default is 26
18     """
19    GPIO.setmode(GPIO.BCM)
20    GPIO.setup(pin,GPIO.OUT)
21    #GPIO.output(pin,False)
22
23
24
25 if __name__ == '__main__':
26     unlock()
27     time.sleep(10)
28     lock()
29     GPIO.cleanup(26)
```

The status bar at the bottom right shows "Python" and a gear icon.

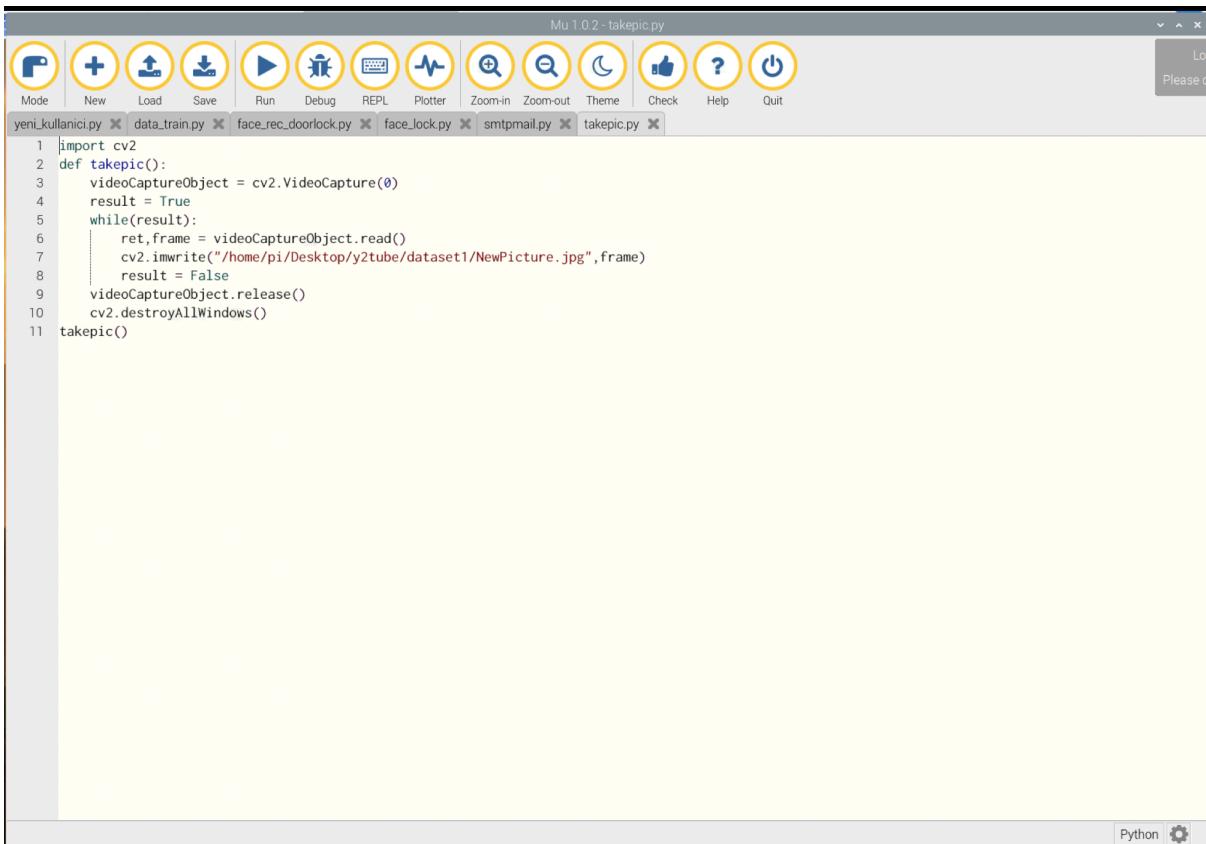
3.1.6.SMTP MAIL GÖNDERME



The screenshot shows the Mu 1.0.2 Python IDE interface. The title bar reads "Mu 1.0.2 - smtpmail.py". The toolbar contains icons for Mode, New, Load, Save, Run, Debug, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar, several tabs are visible: "yeni_kullanici.py", "data_train.py", "face_rec_doorlock.py", "face_lock.py", "smtpmail.py" (the active tab), and "takepic.py". The main code editor area contains the following Python script:

```
1 import smtplib
2 import datetime
3 from email.mime.multipart import MIMEMultipart
4 from email.mime.image import MIMEImage
5 from email.mime.text import MIMEText
6 import os
7 import cv2
8 import takepic as t
9 now = datetime.datetime.now()
10
11 t.takepic()
12
13
14
15 def mails():
16     content="RAPOR"
17     msg=MIMEMultipart()
18     k="someone tried to enter the house on time "+str(now.strftime("%Y-%m-%d %H:%M:%S"))
19     text = MIMEText(k)
20     msg.attach(text)
21     img_data = open("/home/pi/Desktop/y2tube/dataset1/NewPicture.jpg", 'rb').read()
22     image = MIMEImage(img_data, name=os.path.basename("/home/pi/Desktop/y2tube/dataset1/NewPicture.jpg"))
23     msg.attach(image)
24
25     mail=smtplib.SMTP("smtp.gmail.com",587)
26
27     mail.ehlo()
28
29
30     mail.starttls()
31
32
33     mail.login("yagmurrkahya22@gmail.com", "████████")
34     mail.sendmail("yagmurrkahya22@gmail.com", "yagmurrkahyaa@gmail.com",msg.as_string())
35
36 mails()
```

3.1.7. SMTP MAIL GÖNDERİRKEN FOTOĞRAF ÇEKME FONKSİYONU



The screenshot shows the Mu 1.0.2 Python IDE interface. The title bar reads "Mu 1.0.2 - takepic.py". The toolbar contains icons for Mode, New, Load, Save, Run, Debug, REPL, Plotter, Zoom-in, Zoom-out, Theme, Check, Help, and Quit. Below the toolbar, several Python files are listed in tabs: "yeni_kullanici.py", "data_train.py", "face_rec_doorlock.py", "face_lock.py", "smtpmail.py", and "takepic.py". The main code editor window displays the following Python script:

```
1 import cv2
2 def takepic():
3     videoCaptureObject = cv2.VideoCapture(0)
4     result = True
5     while(result):
6         ret,frame = videoCaptureObject.read()
7         cv2.imwrite("/home/pi/Desktop/y2tube/dataset1/NewPicture.jpg",frame)
8         result = False
9     videoCaptureObject.release()
10    cv2.destroyAllWindows()
11 takepic()
```

In the bottom right corner of the code editor, there is a "Python" button with a gear icon.

SONUÇ VE ÖNERİLER

Önerilen model, maliyet olarak pratik biçimde uygulanabilirdir.

Sadece ev için değil iş yerleri içinde kullanılabilir.

Uygulamanın dezavantajı olarak görülebilecek en önemli unsurlar internet bağlantısının kopması veya elektriğin gitmesi olarak düşünülebilir.

İlerideki çalışmalarda ,sistemin geliştirilmesi için çeşitli eklemeler yapılabilir.

Bunlardan bazıları, sistem için bir mobil uygulama geliştirilmesi, uygulama aracılığı ile kilidin tek tuşla kullanıcı tarafından açılabilmesi, kullanıcı kişinin canlı olarak kamera görüntülerine erişebilmesi olabilir.

KAYNAKÇA

- (1) Kalyoncu, A. & Turan M. (2020). IoT Teknolojisi Kullanan Pratik ve Güvenilir Akıllı Kapı Kilidi Tasarımı. Avrupa Bilim ve Teknoloji Dergisi, (Special Issue), 43-49.
- (2) Access Control of Door and Home Security by Raspberry Pi Through Internet,Md. Nasimuzzaman Chowdhury¹ , Md. Shiblee Nooman² , Srijon Sarker³
- (3) WirelessUP projesi ,bolvadinmyo.aku.edu.tr
- (4) www.onbirkod.com/Python ile Mail Göndermenin En Kolay Yolu
- (5) maker.pro/raspberry pi projects
- (6) www.raspberrypi.org
- (7) [electronicshub.org/How to Control a Relay using Raspberry Pi?](http://electronicshub.org/How-to-Control-a-Relay-using-Raspberry-Pi?)
- (8) [elektrikelektronikprojeleri.blogspot.com/Raspberry Pi - GPIO İşlemleri](http://elektrikelektronikprojeleri.blogspot.com/Raspberry-Pi-GPIO-Islemleri)
- (9) pyimagesearch.com/
- (10)<https://hayaletveyap.com/yuz-tanima-algilama-sistemi-yapimi-nasil-yapilir/>
- (11)Python İle Derin Öğrenme-François Chollet
- (12)<https://www.python.org/>