

**T.C.**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

# GÖMÜLÜ SİSTEMLER VİZE

# PROJE RAPORU

**ÖĞRENCİLERİN**

**ADI SOYADI :** FURKAN VARLI, MUHAMMET EMİN AYDINALP

**NUMARASI :** 170215021, 170215020

**PROJE KONUSU**: AKILLI GÜVENLİK KAMERASI

**Akıllı Güvenlik Kamerası**

**Giriş**

Bu projede evlerin güvenliği amaçlı akıllı güvenlik kamerası tasarlanmıştır. Raspberry pi üzerine kurulu olan kamera ve yazılımlar sayesinde hareketi algılayarak açılan, insan tepit ettiğinde fotoğrafını çekip mülk sahibine mail atan bir güvenlik kamerasıdır. Mülk sahibinin evde olmadığı zamanlarda evindeki hareketlilik hakkında kişiye bilgi vermektedir. Proje bilgisayar tabanı olarak Raspberry Pi kullanmakta, kamera (OpenCV yazılımı) yardımıyla hareketin kaynağının insan olması durumunda mülk sahibine mail atmaktadır.

Burada önemli olan nokta OpenCV yazılımı hareketin kaynağının insan olup olmadığını kontol ederek eğer insansa sistemin mail atmasını sağlamaktır. Amaç, insan dışı hareket kaynağı unsurlarının ayrıştırılmasını sağlamak ve gereksiz yere mail atılmasını önlemektir.

**Gerekli Donanım Bileşenleri**

1. 1 adet Raspberry Pi
2. 1 adet Kamera modülü

**Gerekli Yazılım Bileşenleri**

1. Raspbian Jessie OS ([www.raspbian.org](http://www.raspbian.org))
2. OpenCV (https://opencv.org/)
3. Python3

Kullanılan Bileşenlerin Özellikleri

1. Raspberry Pi 3 Model B: Bünyesinde, Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM, 40 pin GPIO, MicroSD port, HDMI girişi bulunduran, üzerinde Linux tabanlı işletim sistemlerinin çalıştırılabildiği mini bilgisayar. (direnc.net sitesinden temin edilebilir)
2. Logitech Web Kamerası: 5 MP çözünürlükte, HD 720p video çekebilen web kamerasıdır. Logitech yetkili bayilerinden veya herhangi bir teknoloji marketinden alınabilir. Özellikleri:

Önerilen sistemde görüntülü arama (640 X 480 piksel)

Görüntü yakalama: En çok 1024 X 768 piksel

Logitech Fluid Crystal™ Teknolojisi

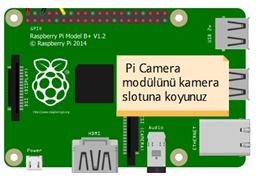
Fotoğraflar: 5 megapiksele kadar (yazılım destekli)

Gürültü azaltma teknolojisine sahip yerleşik mikrofon

Yüksek Hızlı USB 2.0 sertifikalı (önerilen)

Evrensel klips dizüstü bilgisayarlara, LCD veya CRT monitörlere uyar

**Şematik Çizimi**



ŞematikÇizim.jpg

**Yapım Aşamaları**

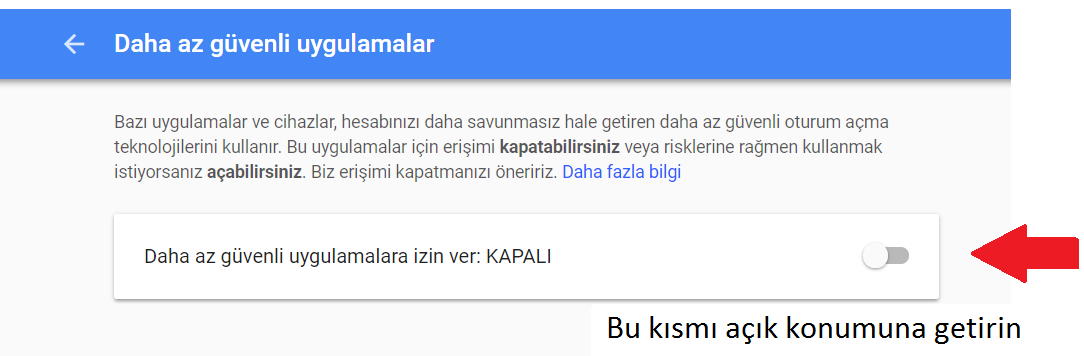
**OpenCV Kurulumu**

<https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/> adresindeki adımlar takip edilerek kurulum yapılabilir.

**Gmailden Daha Az Güvenli Uygulamalar İçin Erişimi Açmak**

Mail Gönderme esnasında kullanıcı adınızı ve şifrenizi doğru girseniz dahi program bir sıkıntıyla karşılaşacaktır. Bunun sebebi ise gmailin mail serverini daha az güvenli uygulamalara kapatmasından kaynaklanmaktadır. Özel olarak aşağıdaki linkten daha az güvenli uygulamalara izin verilmesi gerekmektedir.

<https://myaccount.google.com/lesssecureapps>



GoogleMailGüvenlik.png

**Python Kodu**

|  |
| --- |
| #Kütüphaneleri ekliyoruz  **import** time**,** io**,** picamera**,** cv2**,** numpy**,** smtplib**,** getpass**,** socket**,** datetime  **from** email **import** encoders  **from** datetime **import** datetime  **from** email**.**mime**.**multipart **import** MIMEMultipart  **from** email**.**mime**.**base **import** MIMEBase  **from** email**.**mime**.**text **import** MIMEText  mail\_gonderici **=** "gonderici@maili.com"  mail\_sifresi **=** "gonderici maili sifresi"  kime\_gidecek **=** "alici@maili.com"  baslik **=** "Mail Konusu"  stream **=** io**.**BytesIO**()**  #hareket tespit algoritması  **def** farkImaj**(**t0**,**t1**,**t2**):**  #girilen fotoğrafların mutlak değerle farkını alıyor  # eğer fotoğraflar farklıysa 1 değilse 0 döndürüyor  fark1**=**cv2**.**absdiff**(**t2**,**t1**)**  fark2 **=** cv2**.**absdiff**(**t1**,** t0**)**  **return** cv2**.**bitwise\_and**(**fark1**,**fark2**)**  **def** hareket\_tespiti**():**  # fotoğrafın değişim eşiği belirlendi  esik\_deger**=**200000  #kamera bağlantısı  kamera**=**cv2**.**VideoCapture**(**0**)**  #kameradan alınan görüntüleri gray uzayına çevrilmesi  t\_eksi**=**cv2**.**cvtColor**(**kamera**.**read**()[**1**],**cv2**.**COLOR\_BGR2GRAY**)**  t**=**cv2**.**cvtColor**(**kamera**.**read**()[**1**],**cv2**.**COLOR\_BGR2GRAY**)**  t\_arti**=**cv2**.**cvtColor**(**kamera**.**read**()[**1**],**cv2**.**COLOR\_BGR2GRAY**)**  #zaman kontrolü  zamanKontrol**=**datetime**.**now**().**strftime**(**'%Ss'**)**    **while** **True:**  # eğer fotoğrafların farkı eşik değerinden büyükse ve  # fotoğraflar farklı zamanlarda çekilmişse while döngüsünden çık  **if** cv2**.**countNonZero**(**farkImaj**(**t\_eksi**,**t**,**t\_arti**))>**esik\_deger **and** zamanKontrol **!=**datetime**.**now**().**strftime**(**'%Ss'**):**  **break**  zamanKontrol **=** datetime**.**now**().**strftime**(**'%Ss'**)**    t\_eksi**=**t  t**=**t\_arti  t\_arti**=**cv2**.**cvtColor**(**kamera**.**read**()[**1**],**cv2**.**COLOR\_BGR2GRAY**)**  #while döngüsünden çıkıldığında hareket\_tespiti fonksiyonu true döndürüyor  **return** **True**  # insan tespit algoritması  **def** vucut\_tespiti**():**  #kamera bağlantısı  kamera**=** cv2**.**VideoCapture**(**0**)**  # opencvde insan vücudu tanıma sınıflandırıcısı tanımlandı  body\_cascade **=** cv2**.**CascadeClassifier**(**'/home/pi/opencv-3.4.0/data/haarcascades/haarcascade\_fullbody.xml'**)**  **while** **True:**  #kamera okundu  ret**,** frame**=**kamera**.**read**()**  #görüntü gray uzaya çevrildi  gray**=**cv2**.**cvtColor**(**frame**,**cv2**.**COLOR\_BGR2GRAY**)**  bodys**=**body\_cascade**.**detectMultiScale**(**griton**,**1.3**,**4**)**    **print(**str**(**len**(**bodys**))+**" insan tespit edildi"**)**    #insan vücudunun kutu içine alınması  **for** **(**x**,**y**,**w**,**h**)** **in** bodys**:**  cv2**.**rectangle**(**frame**,(**x**,**y**),(**x**+**w**,**y**+**h**),(**0**,**255**,**255**),**3**)**  # tespit edilen insanın görüntüsünü kaydediyor  cv2**.**imwrite**(**'tespit.jpg'**,**frame**)**  **if** len**(**bodys**)>** 0:  **break** # Döngü insan bulana kadar çalışmaya devam edecek  # tespit edilen insan sayısını döndürüyor  **return** len**(**bodys**)**  #mail fonksiyonu  **def** mail**():**  msg **=** MIMEMultipart**()**  msg**[**'From'**]** **=** mail\_gonderici  msg**[**'To'**]** **=** kime\_gidecek  msg**[**'Subject'**]** **=** baslik  #mail mesajı  msg**.**attach**(**MIMEText**(**"Kapinizda biri tespit edildi!"**,** 'plain'**))**  # tespit edilen insan fotoğrafı ve dosya yolu tanımlanıyor  dosya\_yolu **=** open**(**"tespit.jpg"**,** "rb"**)**  dosya\_adi **=** "tespit.jpg"  part **=** MIMEBase**(**'application'**,** "octet-stream"**)**  part**.**set\_payload**((**dosya\_yolu**).**read**())**  encoders**.**encode\_base64**(**part**)**  part**.**add\_header**(**'Content-Disposition'**,** 'attachment; filename= %s' **%** dosya\_adi**)**  msg**.**attach**(**part**)**  mesaj **=** msg**.**as\_string**()**  server **=** smtplib**.**SMTP**(**"smtp.gmail.com"**,** 587**)**  server**.**ehlo**()**  server**.**starttls**()**  server**.**login**(**mail\_gonderici**,** mail\_sifresi**)**  server**.**sendmail**(**mail\_gonderici**,** kime\_gidecek**,** mesaj**)**  **print(**"E-mail başarıyla gönderildi"**)**  **def** main**():**  **while** **True:**  **if** hareket\_tespiti**():** # eğer hareket tespit edilmişse  **if** vucut\_tespiti**()** **>** 0**:** # eğer tespit edilen insan varsa  mail**()** # mail gönderilir  **if** \_\_name\_\_ **==** '\_\_main\_\_'**:**  main**()** |

**Kaynak Kodu**

Projenin resimlerine ve kaynak kodlarına <https://github.com/furkanvarli/SmartSecurityCamera> adresinden, proje videosuna <https://www.youtube.com/watch?v=PJo6KclOm5s&t=34s> adresinden erişilebilir.

**Nasıl Kullanılır**

Pyton3 dosyası oluşturup yukarıdaki kodları kopyalayıp yapıştırın. Gerekli bağlantıları yapın. OpenCV’yi Raspberry Pi’nize yükledikten sonra terminalinize şu kodu yazıp çalıştırın:

python3 name\_of\_file.py

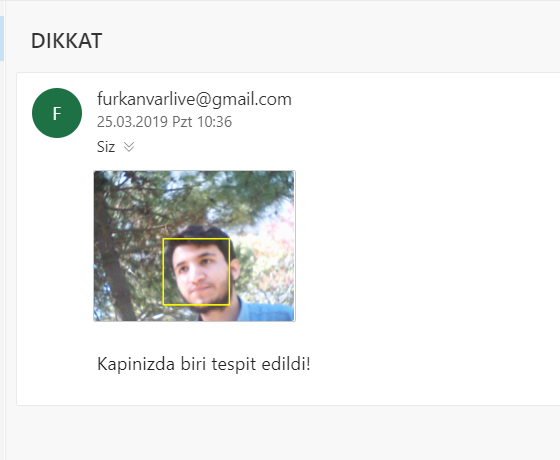
Ardından hareket ettiğiniz anda kamera devreye girecek, eğer insan yüzü varsa fotoğraf çekip belirttiğiniz mail adresinden belirttiğiniz mail adresine mail atacak.

**Proje Resimleri**



Projeresmi.jpg

**Ekran Görüntüleri**



Mail.png

**Öneriler**

* Projeye dış bir kılıf tasarlanıp daha şık bir görüntü elde edilebilir.
* Projede kamera aracılığıyla canlı yayın açılıp, hareket anında bu canlı yayının bağlantı adresi mail gönderilerek kullanıcının, mülkünü canlı şekilde izlemesine olanak sağlanabilir.
* Kullanıcıda internet erişimi bulunmama (ki bu güvenlik açığı yaratır) ihtimaline karşı bir uyarı SMS’i atılabilir.
* Hareket kaynağının kör noktaya kaçma ihtimaline karşı kamera görüntü kaynağını takip edebilir.