



**Engelsiz Gözlük Projesi**

**QOREİQ**

### PROJE ADI : Engelsiz Yaşam

### AMAÇ : Engel Tanımayan Bir Gözlük Hazırlamak

### HEDEF : Görme Konusunda Engel Duyanlara Engelsiz Bir Yaşam Sağlatmak

### PROJEYİ HAZIRLAYAN : Abdulsamed AK

### DANIŞMAN ÖĞRETMEN : Aydın İncekara, Mehmet Avşar

**PROJENİN YAPILIŞI :**

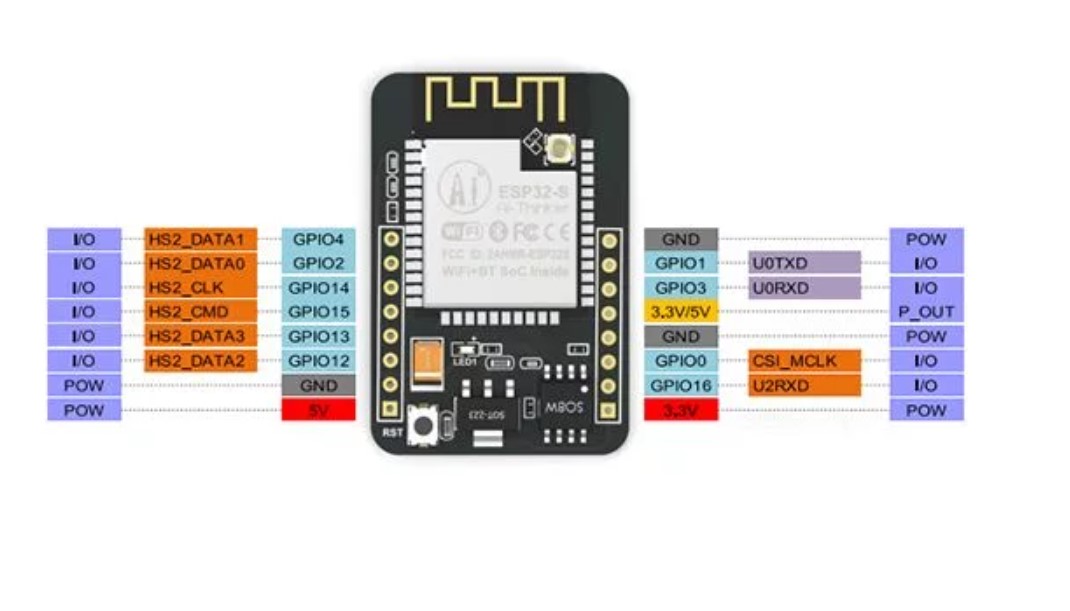
·1.1 **AMAÇ:**

Öncelikle ana amacımız görme engeline sahip kişinin yapmı olacağımız gözlüğü kullanarak tespit edilen insan suretlerinden yola çıkınılarak yapay zeka tarafınca karşısıında kaç adet insan ulunduğunu anlaması, önündeki yazıyı sesli bir şekilde anlaması ve önündeki nesnelere yaklaşınca uyarı alması ayrıca gps tarafınca gideceği yol ie alakalı bilgi alması amaçlanmaktadır.Böyle bir projeyi gerçekleştirmek için kullanmamız gereken özellikler görüntü işleme , mesafe tanıma , sesli asistan gibi özellikler olduğu için malum güçlü bir işlemciye ve haliylede ya ESP32-Cam yada Rasberry Pİ gibi bir karta ihtiyacımız olacaktır.İlk önceliğim gözlüğün tasarımını ayarlamak ve gözlükde kullanacağım sensörlerin kalıbını tasarıma uyarlamak olacaktır.Fakat öncelikle sizlere bu bahsettiğim kartların ne olduğunu, hangisini kullanacağımı, nasıl kullanıldığını ve neler yapılabilir olduğuyla alakalı bilgiler vermek istiyorum.

Bölüm-1 · Modüller|

·1.2 **ESP32CAM MODULU:**

DATASHEET;



Ucuz maliyeti ve düşük güç tüketimi ile Esp32cam birçok projede kendine yer bulabiliyor. Çeşitli varyasyonları, farklı firmalarca üretilip pazarlanan bu mikrokontrolcüler genellikle *Nesnelerin İnterneti (Internet of Things, IoT)* alanında kullanılıyor. Bu modüllerin, yakın bir zamanda geliştirilip piyasaya sürülen bir sürümü kamera içeriyor. ***ESP32-Cam*** olarak adlandırılan bu modül, *OV2640 ve OV7670* kameraları için destek sunuyor. Satın aldığınızda üzerinde genellikle, *OV2640 2MP* bir kamera modülü ile birlikte geliyor.

**ÖZELLİKLER:**

* *Düşük güç tüketimli, çift çekirdekli 32 bit CPU.*
* *240 mhz’e kadar çalışma frekansı, 600 dmips’ye kadar işlem gücü.*
* *Dahili 520 KB SRAM, harici 4 M PSRAM.*
* *UART/SPI/I2C/PWM/ADC/DAC gibi arabirim desteği.*
* *OV2640 ve OV7670 kameralarla çalışabilme, dahili flaş aydınlatma.*
* *Yerel ve Uzaktan (FOTA) yazılım güncelleme desteği*
* *Hafıza kartı desteği*
* *Çeşitli uyku modu seçenekleri*
* *Gömülü Lwip ve FreeRTOS.*
* *STA/AP/STA + AP mod desteği*
* *802.11b/g/n Wi-Fi BT kablosuz desteği*
* *SmartConfig/AirKiss desteği*
* *5V 2A (dahili regülatör bulunuyor)*
* *Boyutlar: 2.7×4 cm/1.06×1.57 inç*

**ESP32CAM İLE NELER YAPILABİLİR?**

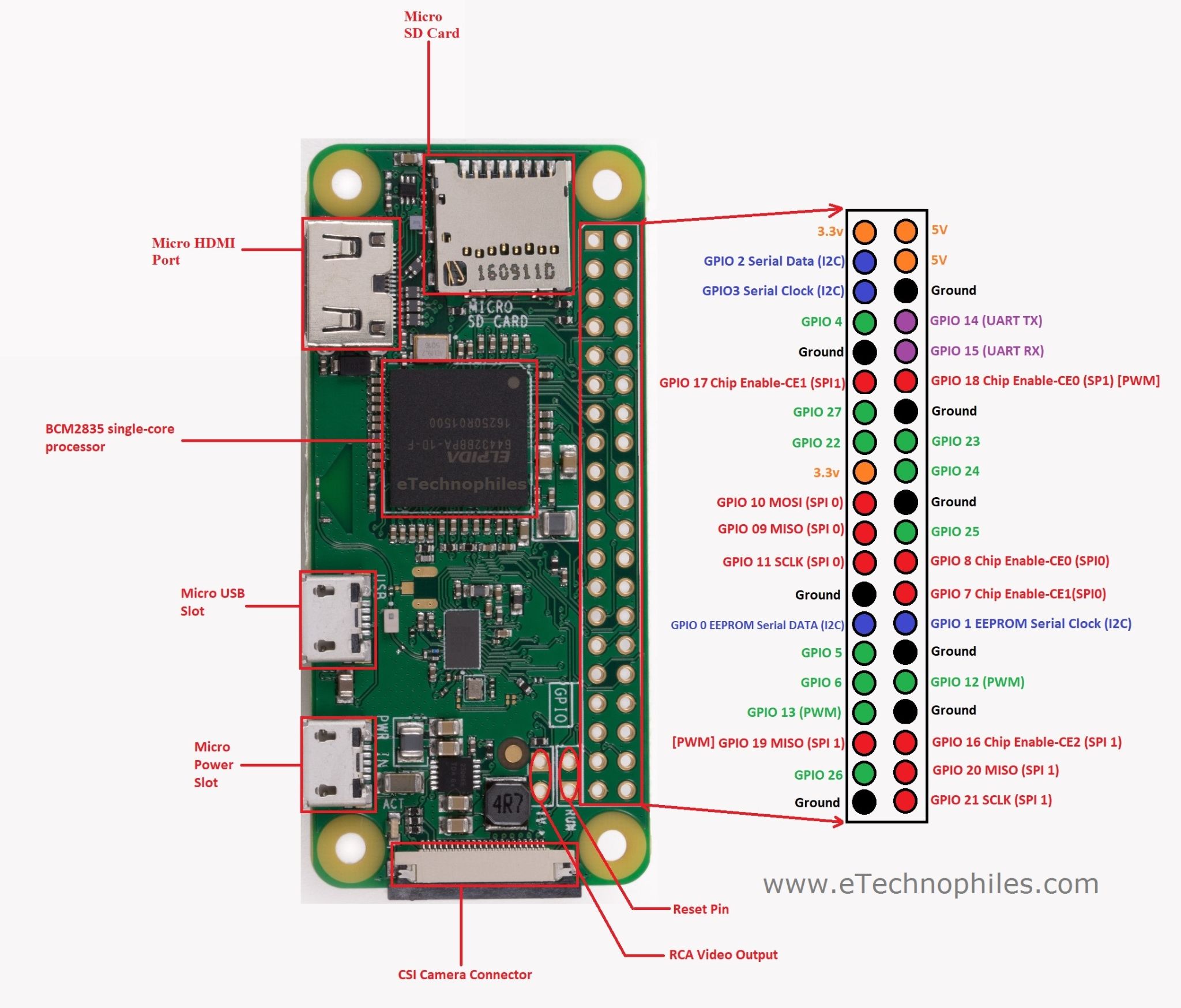
uzaktan kamera görüntüsü almak istediğiniz projeler için hem donanım, hem yazılım desteği ile biçilmiş bir kaftandır. Özgür geliştiriciler projelerini açık kaynak paylaşıyor ve şimdiden bu modül için birçok örnek kodlama, kütüphane internette mevcut durumda. Bu modül ile basit güvenlik projeleri, kablosuz görüntü aktarma, harekete duyarlı fotoğraf çekme, rc araç görüntü aktarımı, qr kod uygulamaları, yüz tanımlama, canlı video aktarımı, fotokapan, görüntülü kapı zili gibi birçok proje, ucuz maliyetle geliştirilebilir.

ESP32CAM [robotistan.com](https://www.robotistan.com/esp32-cam-wifi-bluetooth-gelistirme-karti-ov2640-kamera-modul?language=tr&h=d5f7c5a0&gclid=Cj0KCQjw8IaGBhCHARIsAGIRRYqHyooqANnYO1juHPc51y9DGm8qbTGqc8fxTCHnfhuMtXnzg3NhbtIaAv09EALw_wcB) fiyatıyla **141,91 TL** olmasına rağmen istediğim özelliklere ve mini bilgisayar mantığına sahip bir kart değil.Öncelikle belirli kamera destekleri ve Appinventor’den yazacağım uygulama ile bağlantı konusunda yaşayacağım problemler dolayısıyla tam bu projeye göre bir kart değil.Kullanım alanların görüntü işlemeye dahil olsada güç yetersizliği sebebiyle seçimimi ESP32CAM’dan yapmadım.Görüntü işlemek için daha güçlü bir şeye ihtiyacım vardı ki tahmin edebildiğiniz gibi karşınızda Raspberry Pi!

·1.3 **RASPBERRY Pİ ZERO WH:**

Raspberry Pi serisinin en kullanışlı modellerinden olan Raspberry Pi Zero ile birçok projeyi hayata geçirebilirsiniz. Hem küçük hem de ekonomik bir model olan Raspberry Pi Zero’yu bir SD kart ve orijinal mikro USB çıkışlı bir güç kaynağı ile destekleyerek Linux işletim sistemli bir bilgisayar elde edebilirsiniz. Tek devre kartından oluşan bu mini bilgisayarı; monitör, klavye ve mouse desteği ile gerçek bir bilgisayar gibi kullanabilirsiniz.Bu bilgisayar gibi olma özelliği sebebiyle ise tam benim projemlik bir işlemciye ve tam tasarımıma göre bir boyuta sahiptir.

DATASHEET:;



Raspberry Pi 4 , “”3 “”2 ve “”Zeronun ana ortak noktaları çok fazla projeye elverişli olup boyut olarak malum görseldeki kadar minik cihazlar olmalarıdır ayrıca güç bakımından çok güçlü işlemciye sahip olan Raspberry Pi Zero W’nin kısadan bir nitelikler tablosuna göz atmak istersek:;

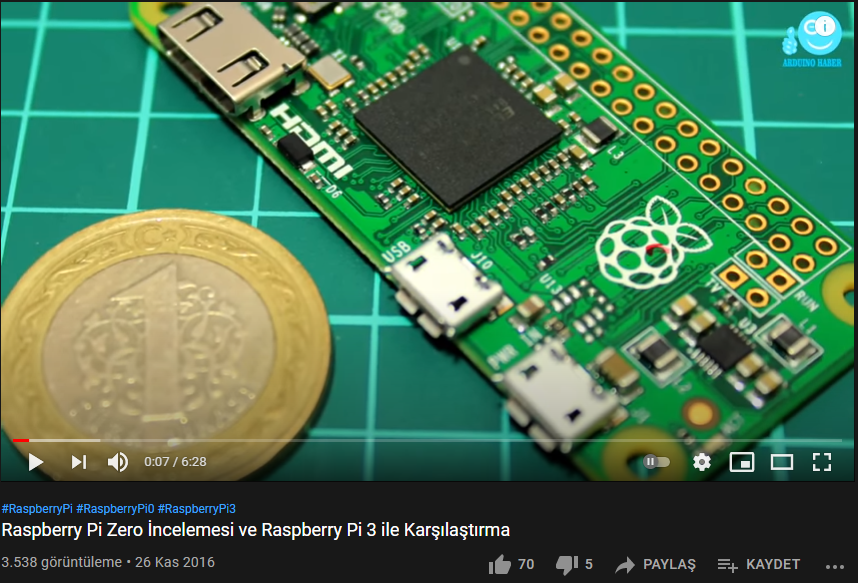
* 1Ghz, Single-core İşlemci
* 512MB RAM LPDDR2
* Ön Lehimli 40 pin Erkek GPIO Header
* Mini HDMI
* USB On-The-Go Giriş
* Micro USB Güç Girişi
* HAT-uyumlu 40-pin header
* Composite video ve reset headerler
* CSI Kamera Konektörü
* 11b/g/n Wireless LAN
* Bluetooth 4.1
* Bluetooth Low Energy (BLE)
* Ağırlık(Kg): 0,5
* Boyut: 6.6 x 3.0 x 0.5 cm

DAHA DETAYLI BİLGİ İÇİN;

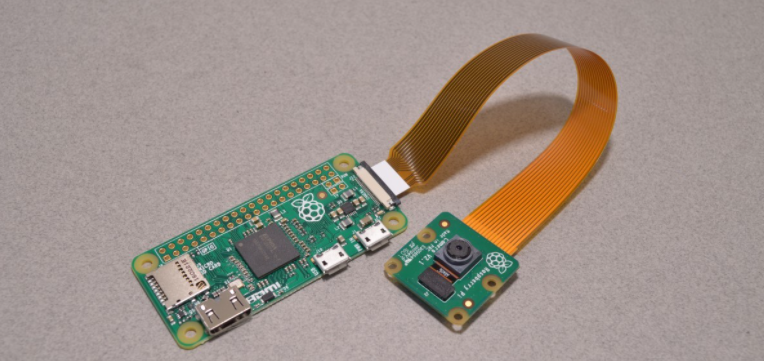
<https://market.samm.com/raspberry-pi-zero-wh>

Ayrıca bknz:

<https://youtu.be/SK7raaBZWik>



Raspberry Pi Zero W’nun kamera girişine sahip olması benim için resmen bir ayrıcalıktı ve kamera konusunda da gene Raspberry Pi ürünü Olan Raspberry Pi Kamera v1 kullanmaya karar verdim.

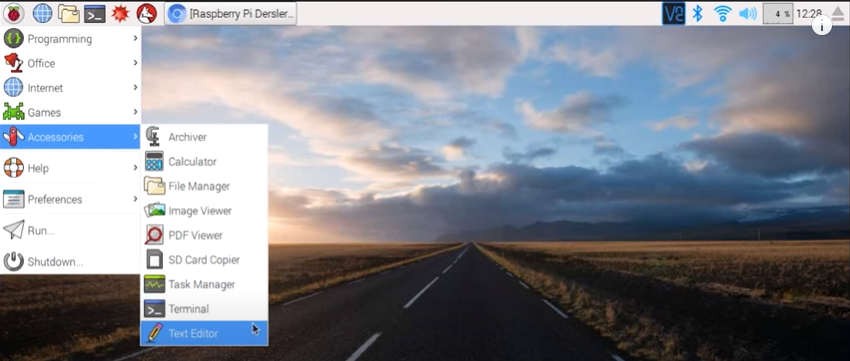


Kamera kablosu Raspberry Pi Zero W’ye göre olmasada bu kablonun dönüştürücüleri var ve bunları kullanarak projeye devam edicem.

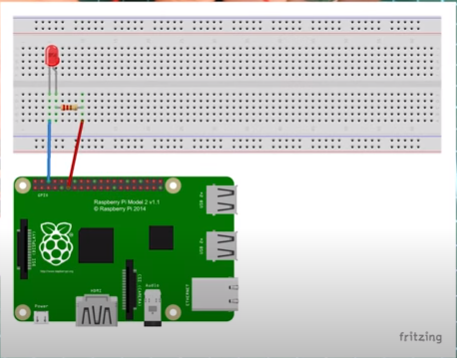
·1.4 **ÖRNEK UYGULAMALAR**

Örnek uygulama bölümü yapmamın sebebi benim anladığım gibi sizinde anlamanız ve kavramanız.Şunu söylemek istiyorum ki Raspberry Pi Zero W üzerinden anlatılan led yakma gibi dersleri bulamamam dolayısıyla robotistann youtube kanalındaki Raspberry Pi 3 çalışmaları üzerinden gitmeyi tercih ettim.

UYGULAMA 1 · RASPBERRY Pİ İLE LED YAKMA| Raspberry pi ile led yakmaya başlamadan önce Raspberry Pi’ye Raspbian işletim sisteminin kurulumunun yapılması gerekmektedir ve bu kurulumun ardından herhangi bir editörden yeni bir metin belgesi oluşturulması gerekmektedir.



Daha sonra mekanik bağlamlarımızı gerçekleştirmemiz gerekecektir ve öncelikle ledimizin kısa bacağını yani gnd “ground” girişini Raspberry Pi üzerindeki 6 numaralı pine bağlayacağız.Sonra ledimizin uzun bacağına led direnci bağlayıp bu direnci 11 numaralı pine bağlamamız gerekecektir fakat Raspberry Pi Üzerindeki GPIO-17 isimli pine bağlamamız lazımdır çünkü raspberry pi üzerinde 11 numaralı pin GPIO-17 isimli pindedir.



Raspberry Pi’deki mekanik kısımların hallolmasının ardından bilgisayarın içerisinde oluşturduğumuz text dosyasının içerisine aşağıda bulunan kodları yaptırmanız gerekmektedir.

import RPi.GPIO as GPIO

import time

# blinking function

def blink(pin):

GPIO.output(pin,GPIO.HIGH)

time.sleep(1)

GPIO.output(pin,GPIO.LOW)

time.sleep(1)

return

# Raspberry Pinin Pin Numaralarını Belirl

GPIO.setmode(GPIO.BOARD)

# set up GPIO output channel

GPIO.setup(11, GPIO.OUT)

# blink GPIO17 50 times

for i in range(0,50):

blink(11)

GPIO.cleanuc()

Yukarıda verdiğim kodları yapıştırdıktan sonra ctrl + shift + s ‘e basarak veya yukarıdan dosya bölümüne gelip farklı kaydet yapmanız ve blink.py olarak kaydetmeniz gerekmektedir.

Bu işlemlerden sonra terminali açıp python python blink.py yazmanız projeyi başlatacaktır ve led 50 kere yanana kadar tekrar tekrar sönüp yanmaya başlayacaktır.

Bölüm-2 · Opencv |

·2.1 **OPENCV NEDİR:;**

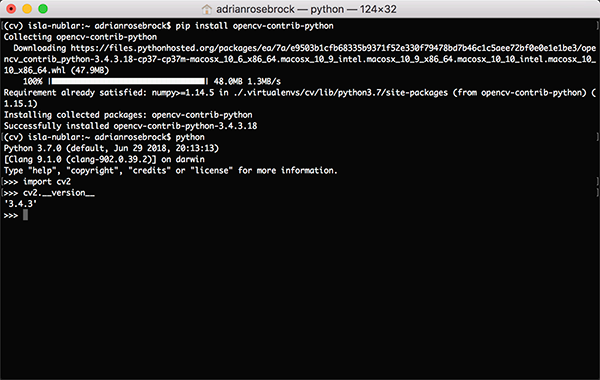
OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500’den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir. Opencv’yi kullanma amacımız insan, haraket ve yazı tespiti gibi işlemlerde kullanmak olduğu için öncelikle bu tanıma işlemlerinin nasıl gerçekleştiğini anlamak gerekmektedir.

·2.2 **KURULUM**

Opencv kurmak için öncelikle python’un kurulu olması gerekmektedir. İstediğiniz python sürümünü python.org’dan indirebilirsiniz. Python’u kurduktan sonra, windovs tuşu + s tuşuna basıp çıkan arama bölümüne cmd yazmalısınız.Karşınıza çıkan siyah pencereli uygulamaya sağ tıklayıp yönetici olarak çalıştır demeniz gerekmektedir.Uygulamamız siyah bir pencere olarak açılmaktadır ve çıkan not defteri gibi ekrana pip install opencv-python yazmanız gerekmektedir.Yazdıktan sonra 2 adet indirme işlemi yapılacaktır ve opencv kurulmuş bulunulacaktır. Bunu kontorl etmek için gene siyah ekrana python yazıp entere basmanız gerekmektedir. Python, cmd üzerinde çalışmaya başlayacaktır ve eğer python paneline import cv2 yazarsanız ve karşınıza hata gelmezse, opencv başarıyla yüklenmiş bulunacaktır. Şimdi sıra aynı ekrana import numpy as np yazmakdır. Bu komutu girdikten sonra da hata ile karşılaşmazsanız, numpy’da başarıyla kurulmuş demektedir.

·2.2 **YAZILIM**

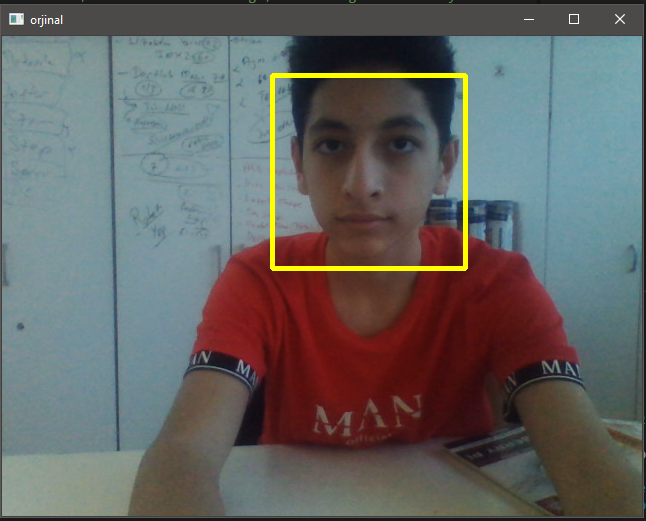
Opencv yazılımsal olarak python üzerinde bir kütüphane olarak çalışır ve görüntü üzerinde çizim gibi işlemlerde yapmamızı sağlar fakat bizim amacımız opencvnin görüntü işleme özelliğini kullanmak ve bu özelliği haarcascade dosyaları üzerinden siyah beyaz resimler üzerinde piksellere göre nesnenin ne olduğunu cihazın algılamasını sağlayacağız. Haarcascadeler demin dediğim gibi siyah beyaz resim veya video üzerinde pixellere göre tanıma yapmamızı sağlayan yardımcı kütüphanelerdir.



Eğer karşınıza böyle bir ekran veya benzeri bir şekilde çıkmış ise opencv başarıyla kurulmuştur.

·2.3 **ÖRNEK ÇALIŞMA**

Örnek olarak yapacığımız çalışma kamera üzerinden ön yüz tanıma ve tanınan ön yüzü kare içerisine alan bir python yazılımıdır



Kodlarımızı yazmak için windows bilgisayar üzerine ister VsCode yardımıyla ister cmd tarafınca python kurulmalı ve sonra yan sayfada verdiğim kodlar yapıştırılmalıdır.

import cv2

import cv2

import numpy as np

kamera = cv2.VideoCapture(0) # kameramızı okutduk

# yuz\_casc diye bir değişkene hardcascade dosyamızı tanıtdık

yuz\_casc = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml")

while(1): # kamera kullandığımız için while döngüsü açdık

ret, frame = kamera.read()

griton = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY) # gri tonumuzu oluşturduk

# yuzler diye bir değişkene detectMultiScale komutu ile hangi resimdeki

yuzler = yuz\_casc.detectMultiScale(griton, 1.1, 4)

# yüzleri bulacağını,%kaç büyüteceğini,kaç kere teyit edeceğini

# söylediğimiz parametreleri ve bu aynı zamanda birden çok

# yüz tanımamızı sağladı

for(x, y, w, h) in yuzler: # for döngüsü ile yüzün etrafında oluşacak karenin konum değişkenlerini girdik ve in yuzler

# diyerek konum değişkelerini yuzler değişkeni içerisinde aldık

cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 255), 3)

cv2.imshow("orjinal", frame)

if cv2.waitKey(25) & 0xFF == ord("q"): # temel kamera kapatma işlemleri

break

kamera.release()

cv2.destroyAllWindows()

Kodları yapışırtırdıkdan sonra internete opencv haarcascade frontal face yazıp github linki bulunmalı yada direk <https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml> bu link üzerinden indirdiğiniz haarcascade soyasını opencv proje klasörünüzün içerisine atmalısınız.Projeyi f5 tuşuna basarak python debug olarak çalıştırdığınızda kameranızdan gelen görüntü açılacaktır ve açılan görüntü üzerinde yüzünüzü sarı renkli kare içerisine alacaktır.

·2.4 **YAŞANABİLECEK SORUNLAR**

·2.4.1 Python debug sorunu: Eğer pythonu bilgisayarınıza kurmadıysanız veya vscode üzerinden kütüphane olarak eklemediyseniz debug açılınılamıyor hatası alırsınız.

·2.4.2 Haarcascade isim sorunu: Eğer haarcascade dosyasını indirdiğiniz yer farklı ise isim değişikliği sebebiyle sorun çıkabilir çözümü ise ya haarcascadeninizin ismini

“haarcascade\_frontalface\_default” yapıcaksınız yada kodlar içerisinden haarcascade olarak tanımlanan isim bölümüne dosya isminizi girmelisiniz.

·2.4.3 Kamera sorunu: Eğer kameranız bilgisayarda dahili bir kamera değilse ayrı olarak takdıysanız kamera = cv2.VideoCapture(0) olan bölümdeki 0’ı 1 yapmanız gerekmektedir.

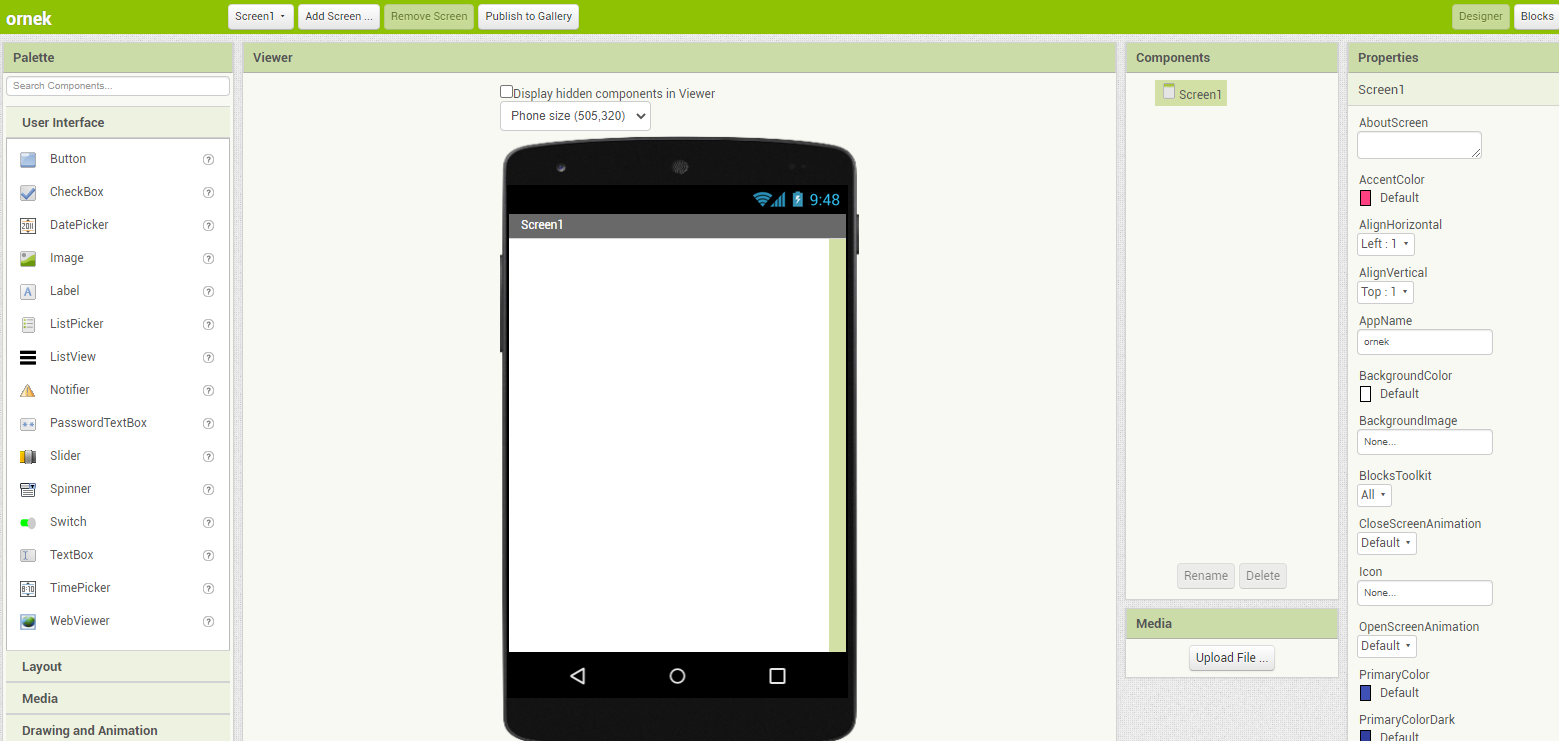
Bölüm-3 · Appinventor |

3.1 **APPİNVENTOR NEDİR**

App Inventor blok tabanlı bir mobil uygulama programlama aracıdır. Her gün daha da gelişen App Inventor ile son derece gelişmiş mobil uygulamalar yapılabilirdir. App Inventor ile şimdilik sadece Android uygulamaları geliştiriliyor. Kullanıcılar uygulamalarını Play Store’a koyarak gelir kazanma ve daha da önemlisi Dünya’ya açılma fırsatı kazanıyorlar. Fakat bizim konumuz bu değil. Asıl amacımız Appinventorun bu uygulama geliştirme arayüzünü ve kütüphanelerini kullanarak görme engellilere google asistan gibi bir yardımcı asistan oluşturmak ve bu oluşturduğumuz yardımcı asistanı raspberry pi ile bağlamaktır.

3.2 KULLANIM

Appinventor kullanabilmek için bir gmail elektronik posta adresinizin olması gerekmektedir.Giriş yapıp uygulamayı açtıktan sonra “Start New Project” seçeneğine tıklayıp önce bir isim verip sonra ok tuşuna tıklıyoruz ve yeni bir uygulama oluşturmuş oluyoruz.



Karşımıza gelen bu ekran üzerinden uygulamamızı tasarlayacağız ve kodlayacağımız uygulamanın tasarımını yaptıktan sonra sağ üstteki “Blocks” butonuna tıklıyoruz ve kodlamamızı ise o tarafda yapıyoruz.

Bölüm-4 · Yapay Zeka|

· 4.1 **YAPAY ZEKA:**

Ne kadarda App İnventoru Tanımış olsakda ortada Raspberry Pi Zero WH ile App İnventorun bağlanıp veri alışverişi yapmalarını sağlamamız gerekmesi gibi bir iğrençlik var yani nedir bu veri alışverişi derseniz de örnek olarak raspberry pi’den giden işlenmiş görüntüden gelen değer App Inventor tarafından sese çevirilip tekrardan kişiye sesli olarak iletilme işlemlerinin tümüne aralarında geçen veri alışverişi diyebiliriz.Bu alışverişin aralarında geçen işlemler ne kadar tatlı dursada bunların hepsi bluetooth ile yapılacak ve internet gerektirecek.İşte İğrençlik diye kast ettiğim kısıma başlıyoruz… Tüm bu işlemlerle uğraşmak yerine App İnventorun yaptığı işlemleri python ilede yapabilmek vardı ki buda zor bir kısım vede bizim 2. Şansımız durumundaydı ki benim terciğim bu kısımdan yana oldu ve şimdi size pythonla nasıl sesli bir şekilde konuşabildiğimiz bir yapay zeka yapacağız.Hadi başlayalım!

· 4.2 **KISACA PYTHON:**

Bildiğimiz python diye projeye başlamak yerine kısaca pythonu anlatmaya karar verdim ve şimdi size pythonu anlatacağım.Kısaca python en popüler yazılım dillerinden biridir ve temel programlama dili işlemlerini (if else, for, while, switch case, değişken atamalar, int, float, array) gerçekleştirebildiğimiz ayrıca bireysel veya şirketleşmiş topluluklar tarafından yazılan kütüphanelerle birlikte projeler geliştirmenizi sağlayan bana göre en kullanışlı yazılım dilidir.Bizde bu kütüphanelerden bikaçı olan speech\_recognition, pyaudio, playsound ve birazdan anlatacağım kütüphaeneler ile birlikte ping pong yani girdi çıktı, etki tepki mantığında çalışan ve bu işlemleri ses ile gerçekleştiren asistan yazılımını yazacağız, tasarlayacağız.

· 4.1 **YAZILIM AŞAMALARI**

Sırasıyla kullanacağımız kütüphaneleri listelemek istiyorum;

import speech\_recognition as sr

from datetime import datetime

import webbrowser

import time

from gtts import gTTS

from playsound import playsound

import random

import os

yukarıdaki bu kütüphaneler sırasıyla sesi yazıya çevirmemizi, internet üzerinden tarihimizi almasını,internet üzerinden arama yapmamızı, saati almamızı, Google aracılığıyla yazıyı sese çevirmemizi, sesi oynatmamızı, rastgele sayı üretmemizi, hem windovs’da hemde linuxda çalışan bi program yazmamızı sağlayan kütüphanelerdir.Kütüphanelerimizi eklediğimize göre diğer kodları yazmaya başlayabiliriz.

r = sr.Recognizer()

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('anlayamadım')

        except sr.RequestError:

            speak('sistem çalışmıyor')

        return voice

def response(voice):

    if 'naber' in voice:

        speak('iyi senden')

    if 'bana bir hikaye anlat' in voice:

        speak('bir varmış bir yokmuş bir adam varmış ölmüş')

    if 'saat kaç' in voice:

        speak(datetime.now().strftime('%H:%M:%S'))

    if 'arama yap' in voice:

        search = record('ne aramak istiyorsun')

        url = 'hhtps://google.com/search?q='+search

        webbrowser.get().open(url)

        speak(search + 'için bulduklarım')

    if 'tamamdır' in voice:

        speak('görüşürüz')

        exit()

def speak(string):

    tts = gTTS(string, lang='tr')

    rand = random.randint(1, 10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

speak('nasıl yardımcı olabilirim')

time.sleep(1)

while 1:

    voice = record()

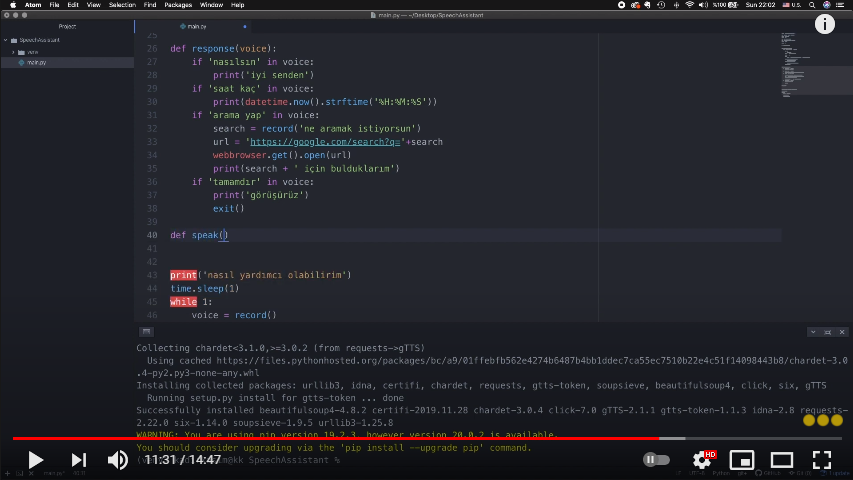
    print(voice)

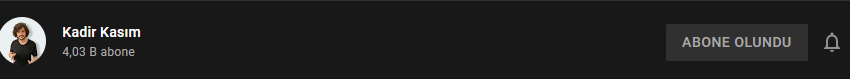
    response(voice)

Gördüğünüz bu kodlar, belirli sorulara cevap veren bir yapay zekayı pythonla yapmış olmamızdı. İstediğimiz sorulara göre cevaplar hazırlayıp hatta opencv projemizi bir değişkene atayıp veyahut direk olarak kopyala yapıştırla bir if içerisine alıp insan sayma komutu oluşturabiliriz ucu çok açık bir proje.

Daha fazla bilgi almanız için size tavsiye etmiş olduğum videoya lütfen uğrayın.bknz;

<https://youtu.be/Nh876KpeXcs>



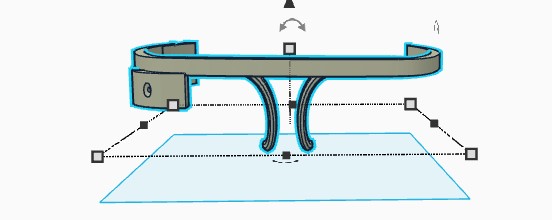


Bölüm-4 · Tasarım|

· 4.1 **TASARIM AŞAMALARI**

Tasarım konusunda internetden bulacağım hazır gözlükler üzerinden tasarımımı ilerletmek istiyordum ve birkaç Sketchup tasarımı bulmuştum.Fakat sorun şuydu ki ben sketch up kullanmasını Tinkercad kadar bilmiyordum vede Tinkercad bana daha rahat geliyordu o yüzden sketchup tasarımlarını stl’ye dönüştürüp Tinkercada import etmeye karar verdim.Vede bir problem daha! Sketch up dosyalarını Tinkercad’e import edince rastgele milimetrik farklar oluşabiliyordu ve bende de oluştu.İnternetden bulduğum farklı tasarımları stl’ye dönüştürüp tinkercada geçirmeyi denedim ve çoğunda sıkıntılarla karşılaşdım.

·4.2 Çok fazla yaptığım tasarım araştırmaları sonucunda çok hoşuma giden bir şeyle karşılaştım. Google’nin 2013 yılında piyasaya sürdüğü hiç tutulmayan gözlüğü Google Glass.Thingiverse’den bu gözlüğün tasarımlarını buldum ve Tinkercad’a import edince milimmetrik bozukluklar yaşamadım ki zaten stl dosyası olarak koymuştu oraya yaptığı tasarımı koyan adam.



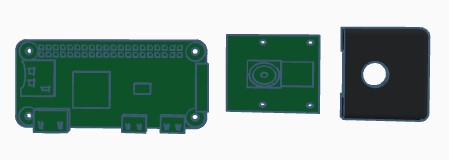
·4.3 Thingiverse’den hazır olarak aldığım Google Glass stl’sini düzenlemek içinkolları sıvadım.Öncelikle nesneyi düzenlemeye başlamak için gözlükten kullanacağım malzemelern bi listesini çıkarmalı, ve onların kalıplarını oluşturmalıydım.Sırasıyla;

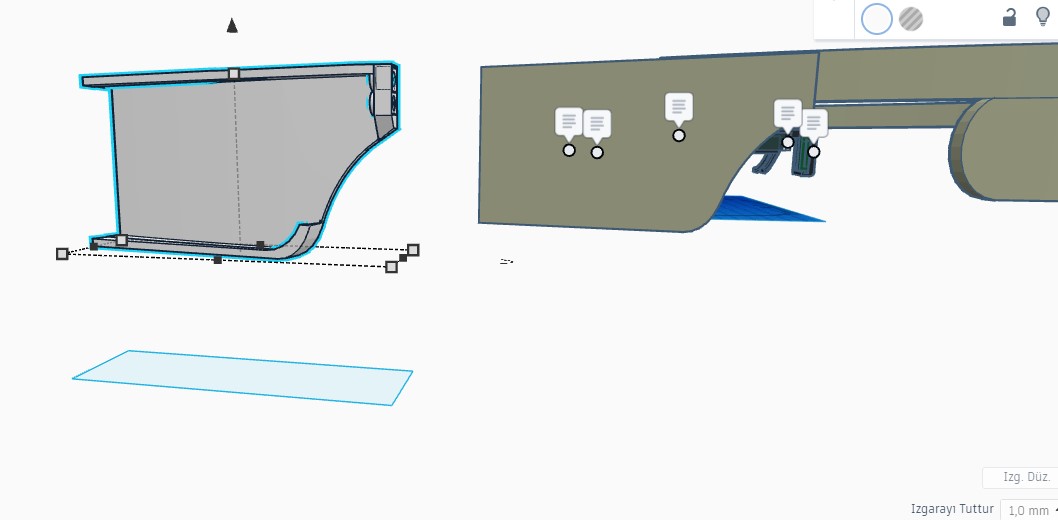
· Raspberry Pi Zero WH ·3.7V 950mah Li-po pil

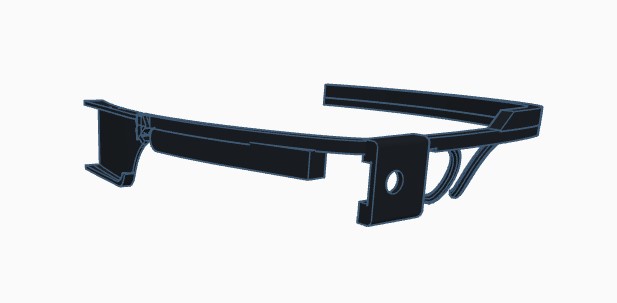
· Raspberry Pi Kamera V1 ·3.7V - 5V Dönüştürücü

· Ultra Sonic Mesafe Sensörü ·Minik Hoparlor

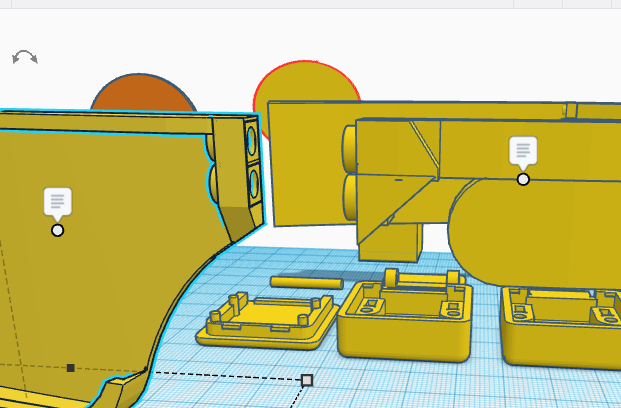
· Şarz Entegresi · Jumper Kablolar

****

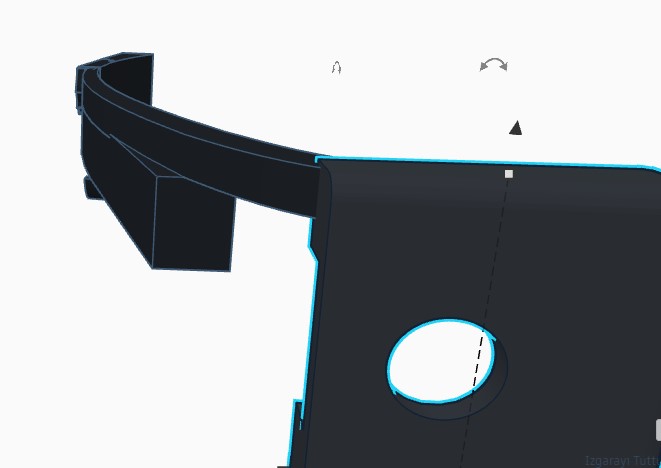
·4.4 Malzemeleri belirledikten sonra kalıpları oluşturmaya başlayacaktım ve öncelikle Raspberry Pi Zero’nun kendisinden başlamaya karar verdim. Öncelikle nun boyutunda bi kutu hazırladım ve sonra vazgeçdim çünkü google glass stlmin içerisinde zaten tam benim istediğim gibi bir kısım vardı fakat tek sorunu boyutuydu ki bunuda raspberry pinin boyutlarına göre ayarlayıp içerisine raspberry pi girecek şekilde bir oyukluk açtım ve aslında sadece vida girişlerini ayarladıktan sonra raspberry pi kalıbımız hazır olacaktı.



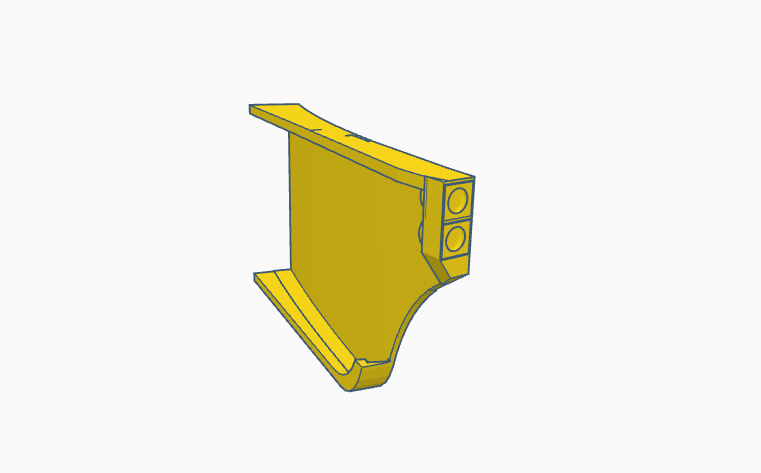
Raspberry Pi Zero WH’nin koyulacağı kalıbın gözlük ile tak çıkar mantığında bağlanması için lego stilinde birleştirgeçler oluşturarak raspberry pi zero kalıbımızı hazırlamıştık.



·4.1 Aslında normalde sırasıyla bakarsak şu an mesafe sensörünün kalıbıyla alakalı bir güncelleme yapmam gerekiyordu fakat mesafe sensörü yerine opencv’nin hareket algılama özelliğini keşfettiğimiz için gerek kalmadı ve onun yerine kabloların geçeceği bölümlerle ilgili birtakım değişiklikler yapmaya karar verdim ve gözlük tasarımını printledikden sonra belirli aletler yardımıyla gözlük tutakaçlarına ince damarlar oluşturup içerisinden kabloları geçireceğim.



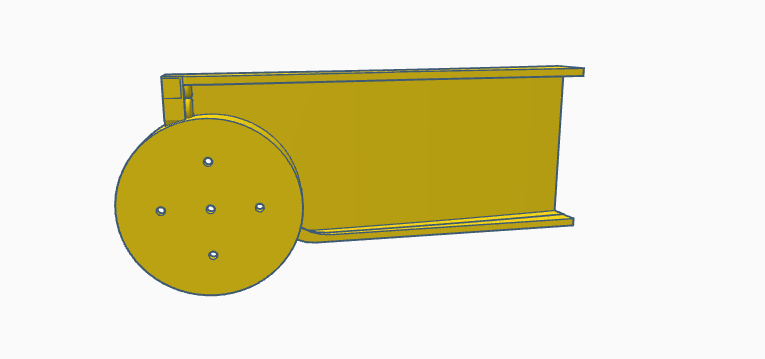
·4.2 Yukarıda gördüğünüz gözlüğün sol tutagacına delici aletler ardımıya kanallar açacağım ve kablo bağlantılarını sağlayacağım. Şimdi ise sıra pil, şarz entegresi, voltaj dönüştürücü ve hoparlörün geçirileceği yeri ayarlamam gerekmekteydi.



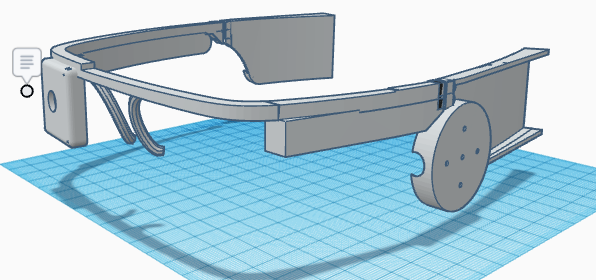
Yapmak istediğim şey Raspberry Pi Zero WH’nin kalıbı olarak oluşturduğum yeri tinkercad’daki ayna aracıyla kalıbı tam tersine çevirmek ve boyutlandırmakdı.Aynı zamanda hoparlör kalıbınıda yapmaktaydım ve hoparlörün boyutunu kumpasla ölçerek ona aşağıdaki gibi bir kalıp hazırladım.



Bu hazırladığım 2 kalıbın son ayarlarını yapıp bi araya getirdiğimde gayet hoş duran ve geniş bir çalışma alanına sahip olmuş oluyorum.



Tasarımın son hali gayet kullanışı ve şık duruyordu.



Bölüm-6 · PROJE|

· 6.1 **PROJEYE BAŞLANGIÇ:**

Projeyi 2’ye ayırıp LITE VERSION ve V2 olarak daha az işlem gerektiren ve çok güçlü ayırt etme yeteneğine sahip 2 versiyon haline getirip Raspberry Pi Zero WH için daha az işlem gerektiren bir versiyonu kurup işlemciyi çok yormadan da aynı projeyi gerçekleştirmeyi sağlamış olacağım.

· 6.2 **LITE VERSION:**

Öncelikle var olan opencv kodları üzerine if else ile kişi sayısını .shape şeklinde zaten numpy’da yani opencv ile gelen işlem kütüphanesinde var olan bir değişken üzerinden printlemeye çalıştım ve sonuç;

    if len(faces) == 0:

        print( "hiç insan yok" ,)

        time.sleep(0.1)

        print( "3 saniye sonra tekrar denenecek" )

        print( "3")

        time.sleep(1)

        print( "2")

        time.sleep(1)

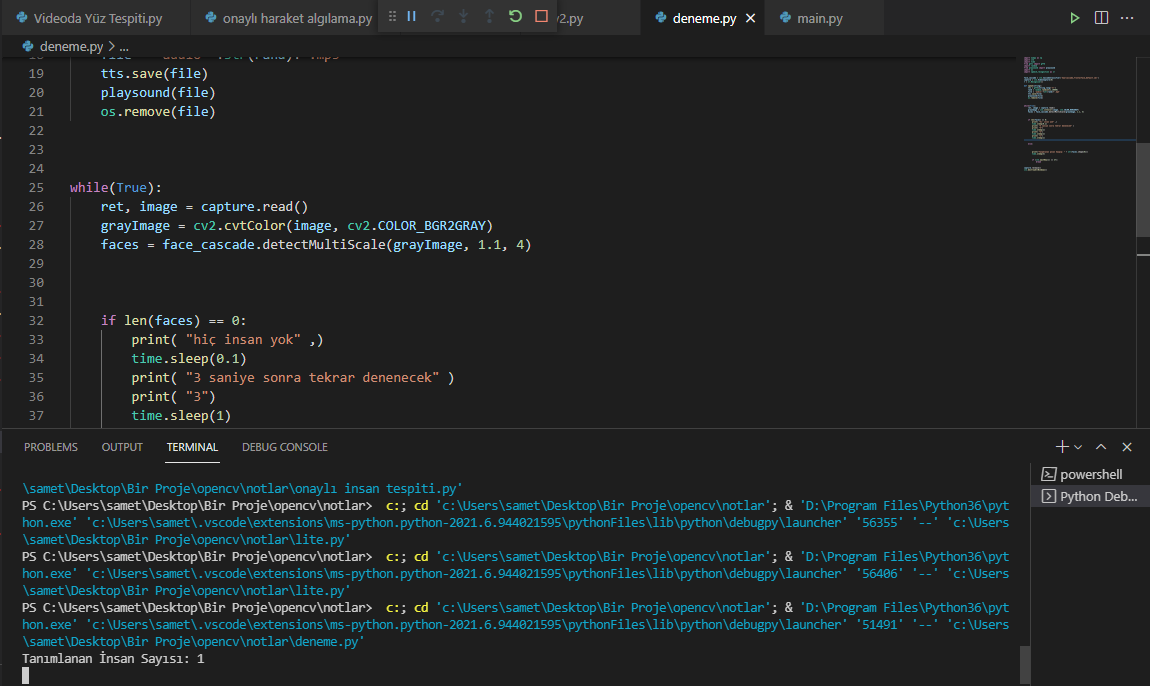
        print( "1")

        time.sleep(1)

    else:

        print("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

        time.sleep(3)



İnsan sayısını printlemeyi başardığımıza göre artık bunları sesli bir şekilde duyma zamanı.Hatırlayacağınız gibi python ile yapay zeka uygulaması bölümünde belirli kütüphaneler yardımıyla text to speech yani yazıyı sese dönüştürmeyi işlemiştik ve bunun için Google’nin gTTS kütüphanesini kullanmıştık şimdi ise bu kütüphaneyi gene bu projeye import edeceğiz ve birkaç satır kod yardımıyla

speak

Diye bir fonksiyon oluşturacağız ve bu fonksiyon print yerine kullanıldığında terminale yazı yazdırmak yerine sesli olarak bir dosya oluşturacak, okutacak ve silecek. Hadi başlayalım

· 6.3 **SESE DÖNÜŞTÜRME:**

Öncelikle imprt edeceğimiz kütüphaneler;

import numpy as np

import cv2

import time

from gtts import gTTS

import random

from playsound import playsound

import os

·time kütüphanesi: “time.sleep” komutu ile çalışmayı saniye bekletmemize yarar.

·gTTS kütüphanesi: Googlenin bu kütüphanesi yazıyı sese dönüştürmemize olanak tanır.

·random kütüphanesi: Rastgele bir sayı üretmemize olanak tanır.

·playsound kütüphanesi: Bir ses dosyasını oynatmamıza olanak tanır.

·os kütüphanesi: Projede kullanma amacımız dosya silme işlemi gerçekleştirmek.

Kütüphanelerin ne işe yaradığını öğrendiğimize göre sese dönüştürmeye başlayabiliriz

Öncelikle “speak” tabirli fonksiyonumuzu oluşturalım ve ne işe yarayacağını belirleyelim

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

Google’nin text to speech kütüphanesini kullanacağımızı söyleyip dilini Türkçe olarak belirledik ve “tts” değişkenini oluşturup ona kaydettik. Sonra random kütüphanesini kullanarak rastgele bir sayı oluşturduk ve “rand” adlı değişkene kaydettik. “file” değişkenimizde sesin kaydolacağı dosya olacaktı ve isminide audio-rand(random sayımız).mp3 olarak ayarladık.playsound kütüphanemiz ile dosyayı çalıştırdık ve sesi bilgisayar hoparlörüne yansıttık.Hatırlarsanız os kütüphanesini dosya silme özelliği için kullanacağımı söylemiştim ve file değişkenimizin sesli olarak okunma işlemi bittikten sonra os kütüphanesini kullanarak silmesini sağladım. Speak fonksiyonumuzu ayarladığımıa göre artık tüm printlerimizi speak fonksiyonumuzla değiştireceğiz yani print kelimesinin bulunduğu yerlere artık speak yazıcağız ve terminale printlenecek tüm yazılar sesli olarak bize gelicek.

Ayrıca, ben yazılım başladığında bana hoş geldiniz denmesini istiyorum bu yüzden, yazılımımızın sonuna

speak('hoş geldiniz')

kodunu ekleyeceğim. Şimdi ise sıra araç sayımı yapmada ve sesli olarak bu araç sayımını söylettirmekte..

· 6.4 **ARAÇ SAYIMI:**

Araç sayımı için ihtiyacımız olan şeyler araba tespit etme haarcascadesi ve belirlenen araçların sayısının speak fonksiyonuyla sesli biçimde belirtilmesi bunun için;

import numpy as np

import cv2

import time

from gtts import gTTS

import random

from playsound import playsound

import os

cap=cv2.VideoCapture(0)

car\_cascade= cv2.CascadeClassfier("cars.xml")

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

while True:

    ret,frame= cap.read()

    arabalar = car\_cascade.detectMultiScale(frame,1.3,3)

    if len(arabalar) == 0:

        speak()

    else:

        speak("Tanımlanan Araç Sayısı: " + str(arabalar.shape[0]))

        time.sleep(3)

        break

cap.relase()

cv2.destroyAllWindos()

Bu şekilde araç sayısını sesli olarak iletebiliyoruz ancak ben insan sayımı gibi insan sayısı 0 demeden sadece araç görürse araç sayısını söyleyen bir yazılım yaptım.Şimdi sıra bu iki yazılımı birleştirmekte.

import numpy as np

import cv2

import time

from gtts import gTTS

import random

from playsound import playsound

import os

car\_cascade = cv2.CascadeClassifier('cars.xml')

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

capture = cv2.VideoCapture(0)

cars= 0

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

while(True):

    ret, frame = capture.read()

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

    faces = face\_cascade.detectMultiScale(grayFrame, 1.1, 4)

    arabalar = car\_cascade.detectMultiScale(frame,1.3,3)

    if len(arabalar) == 0:

        speak()

    else:

        speak("Tanımlanan Araç Sayısı: " + str(arabalar.shape[0]))

        time.sleep(3)

    if len(faces) == 0:

        speak( "hiç insan yok" ,)

        time.sleep(0.1)

        speak( "3 saniye sonra tekrar denenecek" )

        speak( "3")

        time.sleep(1)

        speak( "2")

        time.sleep(1)

        speak( "1")

        time.sleep(1)

    else:

        speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

        time.sleep(3)

        if (cv2.waitKey(1) == 27):

            break

capture.release()

cv2.destroyAllWindows()

Gördüğünüz bu kod iki haarcascadenin tek python dosyasında çalışması için yaptığım birleştirilmiş halidir.

· 6.4 **VERSİON 2:**

V2 yazılımına ekleyeceğim özellikler;

·LITE Versiyondaki Özelliklerin Sesli Komutlarla Çalışması

·Haraket Algılama

·İnternetden Arama Yapma

·Saat bildirme

·Sesli Yapay Danışman

·Yazı Okuma

Tüm bu özellikleri birlikte ekliyeceğiz ve kodlayacağız.Öncelikle, LITE versiyondaki tüm özelliklerin verilen sesli komutlarla çalışmasını sağlayalım.

· 6.4 **SESLİ KOMUTLARA GÖRE İNSAN SAYMA:**

Hatırlıyorsanız python’la yaptığımız yapay zeka uygulamasında aslında kısadan sesli komut alma ve bu komuta göre cevap verme konusuna değinmiştik. Bunun çalışmasını sağlayan şey, python’daki

import speech\_recognition as sr

Kütüphanesi kullanacağız. Bu kütüphane, sesi yazıya dönüştürmemize yarıyor. Bu sayede, kişiden gelen sesi yazıya dönüştürüp gelen yazıya göre insan sayma işlemimizi yani Opencv’deki while döngümüzü çalıştıracağız. Öncelikle cmd’yi (terminaldende olur )yönetici olarak çalıştırıp *pip install SpeechRecognition* yazmanız gerekmektedir ayrıca indirmemiz gereken bir kütüphane daha vardır ve oda *pip install pyaudio* şeklinde indirilebilir fakat bu kütüphanede arkada çalışacak ama import edilmeyecektir. Şimdi, nasıl speak için fonksiyon oluşturduysak ses algılamamızı sağlayan Speech Recognition kütüphanesi içinde birkaç tane fonksiyon oluşturalım.

İlk yapmamız gereken şey kütüphanelerimizi oluşturmak ve değişkenlerimizi ayarlamaktır.

import numpy as np

import cv2

import speech\_recognition as sr

from gtts import gTTS

from playsound import playsound

import random

import os

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

capture = cv2.VideoCapture(0)

r = sr.Recognizer()

Kütüphanelerimizi girdik ve face\_cascade adında bir değişken oluşturup değerini frontalface haarcascademiz yaptık.capture diye oluşturduğumuz değişkeni de kameramız ayarladık.

Bir sonraki adımda ise “r” adıyla bir değişken oluşturuyoruz ve

r = sr.Recognizer()

Değerini viyoruz.Bunu yapmamızdaki amaç aslında ileride kullanacağımız bir kısaltma oluşturmaktır

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

“speak” fonksiyonumuzu oluşturduk.

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('anlayamadım')

        except sr.RequestError:

            speak('sistem çalışmıyor')

        return voice

Burada, “record” fonksiyonumuzu oluşturuyoruz ve *if* ve *try* işlemlerimiz içerisinde “audio” ve “voice” değişkenlerimiz oluşturuyoruz.”voice” değişkenine *r.recognize\_google(audio ,* language=’tr-TR’) değerini veriyoruz ve artık algılama dilimiz yani sesin işlenme dili Türkçe oluyor.

*Expect sr.UnkownValueError:*

*Speak(‘anlayamadım’)* Komutu ile eğer anlaşılmayan bir ses girdisi girilirse “anlayamadım” diye ses çıktısı vermesini istememdir.Aynı şekilde altındaki kodda eğer sistem çalışmıyorsa sistem çalışmıyor diye bir ses çıktısı vermekteyim.Görünüşe göre “record” fonksiyonumuz hazır.Dikkat ettiyseniz birkaç kere de “r” değişkenimizi de kullandım.Hadi bir sonraki aşamaya geçelim.

Şimdi önceki aşamada oluşturduğumuz voice değişkeniyle response türünde bir fonksiyon oluşturacağız ve bunu yapmak için;

def response(voice):

kodunu yazıyoruz ve artık *if* *‘verilecek komut’ in voice:* kullanarak istedğimiz sesi gelen ses içeriside bulup istediğimiz cevabı gene “speak” fonksiyonumuzla verebileceğiz.

if 'insan' in voice:

        while True:

            ret, image = capture.read()

            grayImage = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

            faces = face\_cascade.detectMultiScale(grayImage, 1.1, 4)

            if len(faces) == 0:

                speak( "hiç insan yok" ,)

            else:

                speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

            if (cv2.waitKey(1) == 27):

                    break

        capture.release()

        cv2.destroyAllWindows()

biraz önce anlattığım mantıkta bir if oluşturup içerisine yüz saymak için kullandığımız while döngümüzü koyduk ve *cv2.destroyAllWindows()* ile açılacak olan pencerelerimizi öldürdük ve

speak('nasıl yardımcı olabilirim')

while True:

    voice = record()

    print(voice)

    response(voice)

Burada ise çalıştırma işlemlerimizi ve nasıl yardımcı olabilirim diye başlamasını sağlıyoruz.

İşte Tam Kod;

import numpy as np

import cv2

import speech\_recognition as sr

from gtts import gTTS

from playsound import playsound

import random

import os

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

capture = cv2.VideoCapture(0)

r = sr.Recognizer()

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('anlayamadım')

        except sr.RequestError:

            speak('sistem çalışmıyor')

        return voice

def response(voice):

    if 'insan' in voice:

        while 1:

            ret, image = capture.read()

            grayImage = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

            faces = face\_cascade.detectMultiScale(grayImage, 1.1, 4)

            if len(faces) == 0:

                speak( "hiç insan yok" ,)

            else:

                speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

                    break

        capture.release()

        cv2.destroyAllWindows()

speak('nasıl yardımcı olabilirim')

while 1:

    voice = record()

    print(voice)

    response(voice)

Bu şekilde Verdiğimiz komuta göre insan sayılmasını sağladık.Şimdi sıra konuşabileceğimiz küçük bir yapay asistan yapımında.

· 6.4 **YAPAY ASİSTAN:**

Yapay asistan mı çok zor olsa gerek gibi düşüncelere kapılmanıza gerek yok çünkü gene bi önceki çalışamalarımızda yaptığımız gibi voice fonksiyonumuz içerisinde istediğimiz sesi arayıp cevap hazırlayacağız isterseniz kısadan bi başlayalım öncelikle komut vererek insan sayma yaptığımız kodları alıp üzerinde çalışcağım

import numpy as np

import cv2

import speech\_recognition as sr

from gtts import gTTS

from playsound import playsound

import random

import os

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

capture = cv2.VideoCapture(0)

r = sr.Recognizer()

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('anlayamadım')

        except sr.RequestError:

            speak('sistem çalışmıyor')

        return voice

def response(voice):

    if 'insan' in voice:

        while 1:

            ret, image = capture.read()

            grayImage = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

            faces = face\_cascade.detectMultiScale(grayImage, 1.1, 4)

            if len(faces) == 0:

                speak( "hiç insan yok" ,)

            else:

                speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

            if (cv2.waitKey(1) == 27):

                    break

        capture.release()

        cv2.destroyAllWindows()

speak('nasıl yardımcı olabilirim')

while 1:

    voice = record()

    print(voice)

    response(voice)

Opencv kodlarımızı bitirdiğimiz yerden yani *cv2.destroyAllWindows()* dediğimiz yerden sonrasına

if '' in voice:

Yazıp boş tırnakların içerisine istediğimiz komudu yani örnek olarak

if 'masal anlat' in voice:

    speak('bir varmış bir yokmuş bir adam varmış ölmüş')

Şeklinde komutlar oluşturarak minik yapay zekamızı hazırlamış olacağız.

Hemen arama yap dediğimizde bizim için arama yapmasını sağlayalım

Öncelikle internet üzerinden arma yapması için ihtiyaç duyduğumuz bir kütüphane olan

import webbrowser

Kütüphanesini kullanacağım.

Hadi şimdi temel işlemleri yapalım;

if 'arama yap' in voice:

        search = record('ne aramak istiyorsun')

        url = 'https://google.com/search?q='+search

Şeklinde öncelikle “search” değişkenimizi oluşturup değer olarak içine bizim aramak istediğimiz kelimeyi ayarlayıp “url” değişkenimizin içerisine ”[*https://www.google.com/search?q=*](https://www.google.com/search?q=)*”* + search şeklnde yerleştiriyoruz ve biz ne söylersek “search” değişkenine o kaydediliyor ve ona göre bu değer [*https://www.google.com/search?q=*](https://www.google.com/search?q=)‘ya giriliyor ve googleden bizim için arama yapımış oluyor fakat daha bitmedi çünkü

        webbrowser.get().open(url)

        speak(search + 'için bulduklarım')

Diyerek eklediğimiz kütüphane sayesinde bilgisayar üzerinden oluşturduğumuz linki açıyoruz ve tekrardan “speak” fonksiyonumuzu kullanarak “search” değişkenimizi söyleyip yani aradığımız kelimeyi söyleyip “için bulduklarım” diyor. Bu şekilde de arama yapma işlemimizi yapmış oluyoruz.

Şimdi ise saat kaç dediğimizde saati bize belirtmesini sağlayalım.

Öncelikle kullanacağıız kütüphane

from datetime import datetime

Çünkü internet aracılığıyla saatin kaç olduğunu sürekli öğrenmemiz gerekiyor bu kütüphanemizi dahil ettiğimize göre hadi komutlarımızı ayarlayalım

    if 'saat kaç' in voice:

        speak(datetime.now().strftime('%H:%M:%S'))

Şeklinde datetime kütüphanesi aracılığıyla saat verilerimizi sırasıyla saat dakika ve saniye olarak sesli biçimde belirtiyoruz. Artık sesli olarak tamamdır dediğimizde bizi dinlemeyi bırakmasını sağlayalım fakat önce bizi dinlemeyi başlatması için tüm işlemleri *if* alacağım

def response(voice):

    if 'hey asistan' in voice:

        if 'insan' in voice:

            while 1:

                ret, image = capture.read()

                grayImage = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

                faces = face\_cascade.detectMultiScale(grayImage, 1.1, 4)

                if len(faces) == 0:

                    speak( "hiç insan yok" ,)

                else:

                    speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

                    time.sleep(3)

Gördüğünüz gibi artık hey asistan dediğimde dinleme işlemlerini başlatıcak ve

        if 'tamamdır' in voice:

            speak('görüşürüz')

            exit()

Bu şekilde tamamdır dersem görüşürüz diyip *exit()* yaptıracak.

Artık yapay zekamız kısmen hazır hadi şimdi ara sıcak kıvamında pekiştirme amaçlı yapay arkadaş uygulaması yapalım

· 6.4 **HEY ARKADAŞIM!:**

Şimdi pekiştirme projesi olarak yapay arkadaş projemize başlıyacağız ilk olarak temel kodlarımızı ayarlayalım;

import numpy as np

import cv2

import speech\_recognition as sr

import time

from gtts import gTTS

from playsound import playsound

import random

import os

r = sr.Recognizer()

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('kanka sesin gelmiyor')

        except sr.RequestError:

            speak('beynim yandı knk')

        return voice

def response(voice):

    if 'hey arkadaşım' in voice:

        if 'görüşürüz' in voice:

            speak('görüşürüz kanka')

            exit()

        speak('nasıl yardımcı olabilirim')

        while 1:

            voice = record()

            print(voice)

            response(voice)

Bu şekilde temel kodlarımızı ayrladıktan sonra birinci yapacağımız şey olan biz “hey arkadaşım” diyerek sistemi çalıştırdıktan sonra arkadaşımızın bize “efendim kanka ha buarada taş kağıt makas oynayalım mı” demesini ve bizimde taş kağıt makas uygulaması kodlamamızı istiyorum. Ozaman hadi kodlara geçelim.

    if 'hey arkadaşım' in voice:

        soru = record('efendim knk ha buarada taş kağıt makas oynayalım mı')

        if soru == 'tamam':

            secenek=["taş","kağıt","makas"]

            tas = secenek[0]

            kagit = secenek[1]

            makas = secenek[2]

            arkadasin\_secimi=random.choice(secenek)

seçim = record('taş mı kağıt mı makas mı unutma seni göremiyorum')

            if seçim == 'taş':

                if arkadasin\_secimi == tas:

                    speak('Bende taş demiştim berabere kaldık')

                elif arkadasin\_secimi == kagit:

                    speak('ha ha kanka ben kağıt yapmıştım eziyorum seni he')

                elif arkadasin\_secimi == makas:

                    speak('vallaha ballı çıktın ben makas yapmışdım')

            if seçim == 'kağıt':

                if arkadasin\_secimi == kagit:

                    speak('Bende kağıt demiştim berabere kaldık')

                elif arkadasin\_secimi == makas:

                    speak('ha ha kanka ben makas yapmıştım eziyorum seni he')

                elif arkadasin\_secimi == tas:

                    speak('vallaha ballı çıktın ben taş yapmıştım')

            if seçim == 'makas':

                if arkadasin\_secimi == makas:

                    speak('Bende makas demiştim berabere kaldık')

                elif arkadasin\_secimi == tas:

                    speak('ha ha kanka ben taş yapmıştım eziyorum seni he')

                elif arkadasin\_secimi == kagit:

                    speak('vallaha ballı çıktın ben kağıt yapmışdım')

            speak('ben sıkıldım hadi bay bay')

            exit()

        if soru == 'hayır':

            speak('tamamdır')

        if 'görüşürüz' in voice:

            speak('görüşürüz kanka')

            exit()

Bunlar arkadaşımızla oynadığımız taş kağıt makas oyununun kodları sırayla şu şekil işliyor biz “hey arkadaşım” diyerek sistemi başlatdığımızda o bize “efendim kanka ha buarada taş kağıt makas oynayalım mı” diyerek soru soruyor bu soru içerisinde soru sorma işlemini arama yapma bölümünde görmüştük ve sonra python ile içerisinde taş kağıt makas içeren bi liste oluşturuyoruz sonrasında *tas=secim[0] kagit=secim[1] tas=makas[2]* şeklinde liste elemanlarımızı değişkenlere atıyoruz. Arkadaşımızın seçimini random kütüphanesiyle rastgele seçip ”Seçim” adında bir *record* daha oluşturuyoruz ve sırayla kullanıcının seçimi ile arkadaşın seçimi sürekli karşılaştırılıp oyunun kurallarına göre arkadaş kaybettiğimizi berabere kaldığımızı veya kazandığımızı söylüyor.

Eğer sizinle birlikte yaptığımız uygulamaları uygulayarak bu sayfalara kadar geldiyseniz artık sizde projenin son sayfasında aynı projenin daha iyisini veya daha farklısını belkide dahada gelişmişini yapabileceksiniz size ara ara böyle bonus uygulamalar ve alıştırmalar yapma sebebim daha iyi kaframanızdır.Pekala şimdi projemize bir özellik daha ekleyelim ve artık gözlük bir araç gördüğünde bir tepki göstersin.

Öncelikle Araç tespiti için eklememiz gereken cars.xml haarcascadesine

<https://raw.githubusercontent.com/andrewssobral/vehicle_detection_haarcascades/master/cars.xml>

Üzerinden ulaşabilirsiniz.

car\_cascade = cv2.CascadeClassifier('cars.xml')

“car\_cascade” değişkenini oluşturup haarcascademi tanıttım.

arabalar = car\_cascade.detectMultiScale(frame,1.3,3)

While döngüsünün içerisine tespit etme kodu olan yukarıdaki kodu yazdım.ve bir sonraki satıra ise

if len(arabalar) == 0:

        speak()

    else:

        speak("Tanımlanan Araç Sayısı: " + str(arabalar.shape[0]))

        time.sleep(3)

Komutunu yazıp tanımlanan araç sayısının sesli biçiminde kullanıcıya iletilmesini sağladım.İşte tüm kodlar

import numpy as np

import cv2

import speech\_recognition as sr

from datetime import datetime

import webbrowser

import time

from gtts import gTTS

from playsound import playsound

import random

import os

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

capture = cv2.VideoCapture(0)

car\_cascade= cv2.CascadeClassfier("cars.xml")

r = sr.Recognizer()

def speak(string):

    tts = gTTS(string,lang='tr')

    rand = random.randint(1,10000)

    file = 'audio-'+str(rand)+'.mp3'

    tts.save(file)

    playsound(file)

    os.remove(file)

def record(ask=False):

    with sr.Microphone() as source:

        if ask:

            speak(ask)

        audio = r.listen(source)

        voice = ''

        try:

            voice = r.recognize\_google(audio , language='tr-TR')

        except sr.UnknownValueError:

            speak('anlayamadım')

        except sr.RequestError:

            speak('sistem çalışmıyor')

        return voice

def response(voice):

    if 'hey asistan' in voice:

        if 'insan say' in voice:

            while 1:

                ret, image = capture.read()

                grayImage = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

                faces = face\_cascade.detectMultiScale(grayImage, 1.1, 4)

                if len(faces) == 0:

                    speak( "hiç insan yok" ,)

                    time.sleep(0.1)

                    speak( "3 saniye sonra tekrar denenecek" )

                    speak( "3")

                    time.sleep(1)

                    speak( "2")

                    time.sleep(1)

                    speak( "1")

                    time.sleep(1)

                else:

                    speak("Tanımlanan İnsan Sayısı: " + str(faces.shape[0]))

                    time.sleep(3)

                if (cv2.waitKey(1) == 27):

                        break

            capture.release()

            cv2.destroyAllWindows()

        if 'arac say' in voice:

            while True:

                ret,frame= capture.read()

                arabalar = car\_cascade.detectMultiScale(frame,1.3,3)

                if len(arabalar) == 0:

                    speak()

                else:

                    speak("Tanımlanan Araç Sayısı: " + str(arabalar.shape[0]))

                    break

        if 'bana bir hikaye anlat' in voice:

            speak('bir varmış bir yokmuş bir adam varmış ölmüş')

        if 'saat kaç' in voice:

            speak(datetime.now().strftime('%H:%M:%S'))

        if 'arama yap' in voice:

            search = record('ne aramak istiyorsun')

            url = 'https://google.com/search?q='+search

            webbrowser.get().open(url)

            speak(search + 'için bulduklarım')

        if 'tamamdır' in voice:

            speak('görüşürüz')

            exit()

speak('nasıl yardımcı olabilirim')

while 1:

    voice = record()

    print(voice)

    response(voice)

Gördüğünüz gibi araç tespit etme kodlarınıda bitirdik. Hadi şimdi hareket algılama işlemlerine geçelim.

· 6.4 **HAREKET ALGILAMA:**

Hareket tespiti aslında çok basit ve kavranılabilir bir çalışma stiline sahip etraftaki köntüre ve blura göre hareket var veya yok gibi bi sonuç çıkartıyoruz örnek kod olarak

import cv2

import numpy as np

import time

cap = cv2.VideoCapture(0)

fourcc = cv2.VideoWriter\_fourcc('X','V','I','D')

out = cv2.VideoWriter("output.avi", fourcc, 5.0, (1280,720))

ret, frame1 = cap.read()

ret, frame2 = cap.read()

print(frame1.shape)

while cap.isOpened():

    diff = cv2.absdiff(frame1, frame2)

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

    blur = cv2.GaussianBlur(gray, (5,5), 0)

    \_, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH\_BINARY)

    dilated = cv2.dilate(thresh, None, iterations=3)

    contours, \_ = cv2.findContours(dilated, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

    for contour in contours:

        (x, y, w, h) = cv2.boundingRect(contour)

        if cv2.contourArea(contour) < 900:

            continue

        print("hareket var")

        time.sleep(3)

    #cv2.drawContours(frame1, contours, -1, (0, 255, 0), 2)

    frame1 = frame2

    ret, frame2 = cap.read()

    if cv2.waitKey(40) == 27:

        break

cv2.destroyAllWindows()

cap.release()

out.release()

Şeklinde hareket tespit edebiliyoruz. Opencv’nin kendi eventlerini kullanarak hareket algılama işlemimizi tamamlıyabiliyoruz.

# Sözü Geçen Çalışmalar

<Robotistan.com>

[Direnc.net](direnc.net)

<youtube.com/watch?v=Nh876KpeXcs&t=142s>|Kadir Kasım

[youtube.com/watch?v=hRzBtQh6wuU&list=PLzcys7whQ6eSJdSHJh-xdOLvWPP24J76x](https://www.youtube.com/watch?v=hRzBtQh6wuU&list=PLzcys7whQ6eSJdSHJh-xdOLvWPP24J76x)

<http://ibrahimdelibasoglu.blogspot.com/2016/08/python-opencv-ile-kisi-sayma.html>

<https://jn7.net/python-ve-opencv-ile-yuz-algilama-ve-sayma-uygulamasi/>

