TOBB UNIVERSİTY OF ECONOMICS AND TECHNOLOGY

ELE 495 SENIOR DESIGN PROJECT

Eye Blinking Communcation Project **Final Report**		
Batu Kaan Özen		**141201079**
Date :	16.06.2020	

CONTENTS

1. Abstract...3

2. Introduction...3

3. Project Activities..4

4. Technical Specification...5

5. Budget..8

6. Revised time plan..9

7. References..10

8. Appendix..11

1.) Abstract

Amyotrophic Lateral Sclerosis (ALS), known as motor neuron disease, is a disease that does not impair the mental functions of the patient, but grdually loses muscle control. However, the eyelids of the patient do not lose their function for a long time. For this reason, blink communcation is a recommended solution for ALS patients. Our aim in this project is making one system, which can detect ten words from the number of blinking eye. Our system uses the facial landmarks and their distance between each other and detects eyeblinking after that it dedice the meaning of eye command from the number of blinks. As a example, when we blick 4 times in 10 seconds. It means that the als patient wants banana.

2.) Literature Analysis

The area of computer vision is first started in 1960s aiming to make small amount of image analysis. Before this application many applications are applied manualy such as there were not any system, which can analysis x-rayses. In the beginning of this process, the computer vision algorithms are not satisfactory. As computing power increased, algorithm started to solve individual tast. Until the development in deep learning, the improvement rate is not enourmous but after application with deep learning , the improvement rate in computer vision is improvent signifficantly.

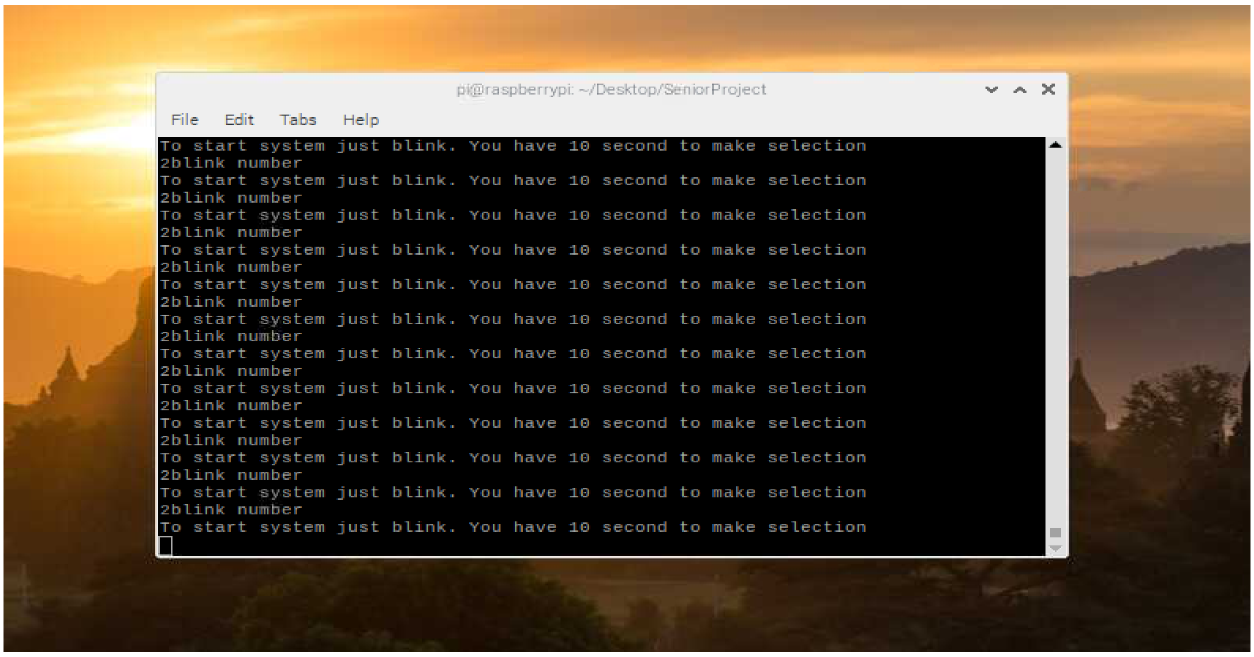
As you know, there are a lot of tool with computer vision system for example, self driving car, radar systems and Infraded detection. The main aim of all these systems is improving human life quality.[1]

One of the important application in computer vision is eyeblinking detector**.** Eyeblinking detection is very important topics in the area of computer vision systems. It is based on main image processing operation. Eye Blinking detection has a great number of appications in an area human-computer interaction such as eye typing applications, mouse control application, google glass. The area is blinking communcation is important topic and there are not suffient research and projects. Because of these reasons, it is very relevant. Even tough, there are significant amount of sucessfull research projects. Such as, Eye Blink Detection for Smart Glasses by Haang Le, Thanh Dang and Feng Liu. The smart glasses project was aiming to construct one blink detection. [2]. The project, Automated Eye Blink Detection and Tracking Using Template Matching, is applied temple matching method and solved task via template matching method.

3.) Project Activities

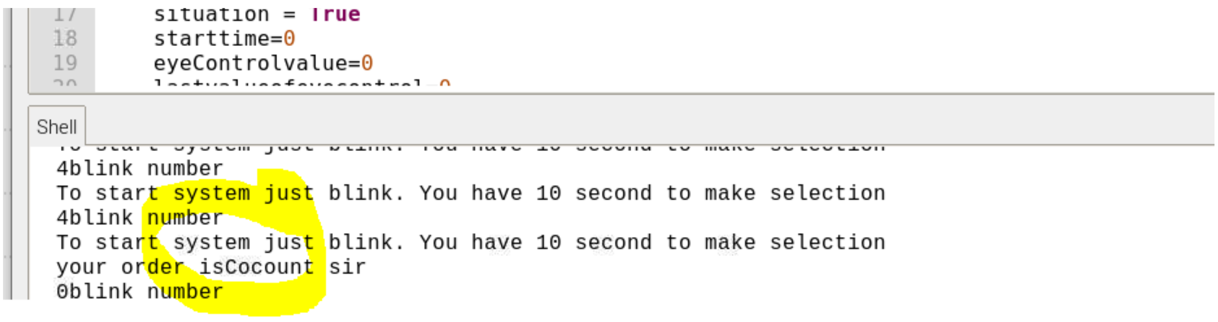
In our first design, it is planned to use Raspberry Pi 3b plus, Raspberry Camera V2.1 and Raspberry Pi 16x2 LCD, SunDisk(32 GB) and Mikro USB charger and we were planning to apply the algorithm, which is described in Technical Spesification part. While the design of senior project one problem is occured, the code running on Linux PC did not work due to the lack of computitional power. Because of this situation, Microcontroller is changed with Raspberry Pi 4B 4GB ram. But there was a still speed problem, so it is changed the resolution of Raspberry Camera V2.1 and code is worked, as aspected. After changing our Microprocessor with Raspberry Pi 4 b 4GB RAM. It was planned to used 2x16 LCD but using 3.5 Inch Lcd Screen was a better solution because of nice wiev. Because of this reason, LCD 3.5 Inch Lcd Screen is used.

You can see our system output in Figure 3.1 and Figure 3.2



```
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
2blink number
To start system just blink. You have 10 second to make selection
```

Figure 3.1



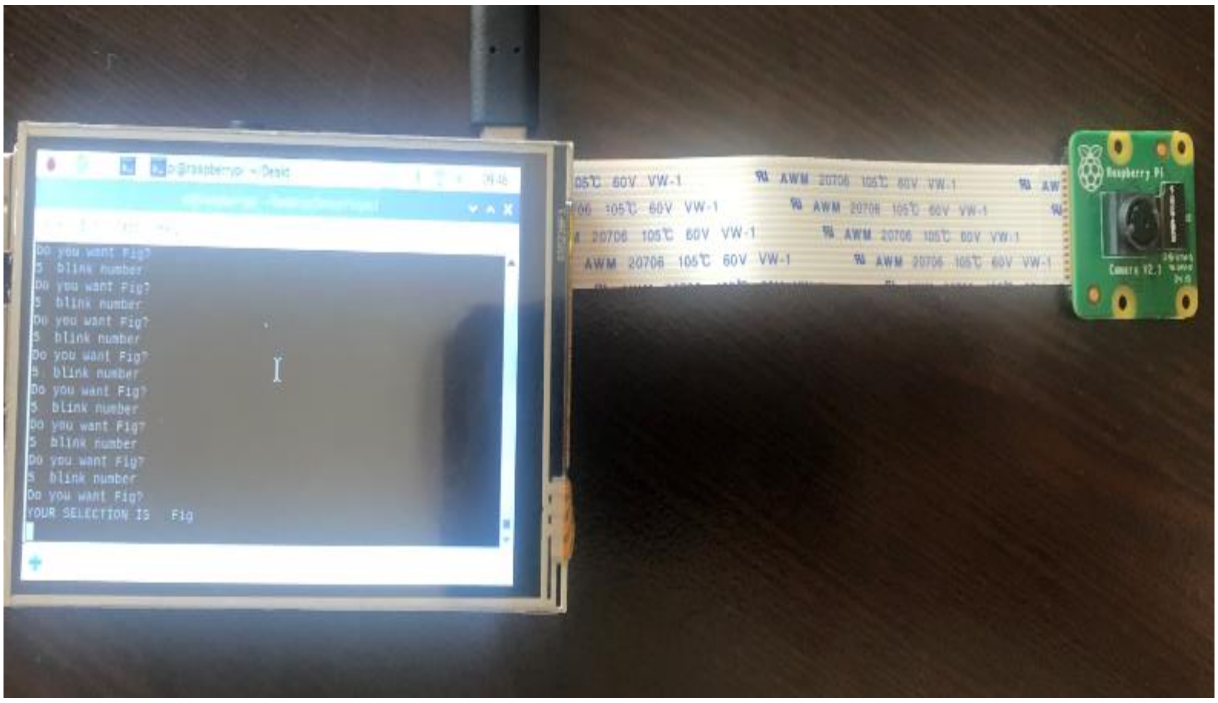
```
    situation = True
    starttime=0
    eyeControlvalue=0
```

Shell

```
4blink number
To start system just blink. You have 10 second to make selection
4blink number
To start system just blink. You have 10 second to make selection
your order isCocount sir
0blink number
```

Figure 3.2

In figure 3.3, you can see our final product and its output.



```
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
5  blink number
Do you want Fig?
YOUR SELECTION IS   Fig
```

Figure 3.3

4.) Technical Specifications

a.) Hardware Design :

In my project, it is planned to using Rasbery Pi 4 b plus with 4GB RAM, Raspberry Camera V2.1, Raspberry Pi LCD Screen , SunDisk (32 GB) and TYPE-C charger .

Power requirement of the Raspberry pi b is 5V/3 A power and it has 1.5 GHZ 64-bit quad-core processor, dual-band wireless LAN, Bluetoothth 4.2/BLE, fast Ethernet and power over Ethernet support withe separete PoE HaT, extended 40-pin GPIO header, Full-size HDMI, 4 USB 2.0 ports, CSI camera port for connecting a Raspberry Pi camera, DSI display port for connecting a Raspberry Pi touchschreen display, 4- pole streo output and composite system and stroing data.[3].

Camera ware v2 modul spesification. Still resolution is 8 Megapixels, video modes 1080p30, 720p60 and 480p60/90. For intergrating to Linux, V4L2 driver is avaliabl. The camera modul has Sony IMX219 sensor and sensor resolution is 3280x2464 pixels.[4]

3.5 TFT LCD 320x480 Touch Display for Raspberry Pi has 320x480 resolution screen and TFT resistive touch screen.

b.) Software Design:

In our system, Raspbian operating system is installed on Raspberry Pi micro controller and our system is planned to find facial landmarks and finding eye blinking movement from facial landmarks. When we are finding facial landmarks, we used Dlib library to detect facial landmarks. In the tast of facial landmarks, our goal is detect facial structures on the face using shape prediction methods. Dlib library applied pre-trained HOG + Linear SVM object detector for the task of face detection. The pre-trained facial landmark detector inside the dlib library is used to estimate the location of 68 (x,y)-cordinates that map to facial structures on the face. You can see dlib face detection landmark in figure 4.b.1.[3]

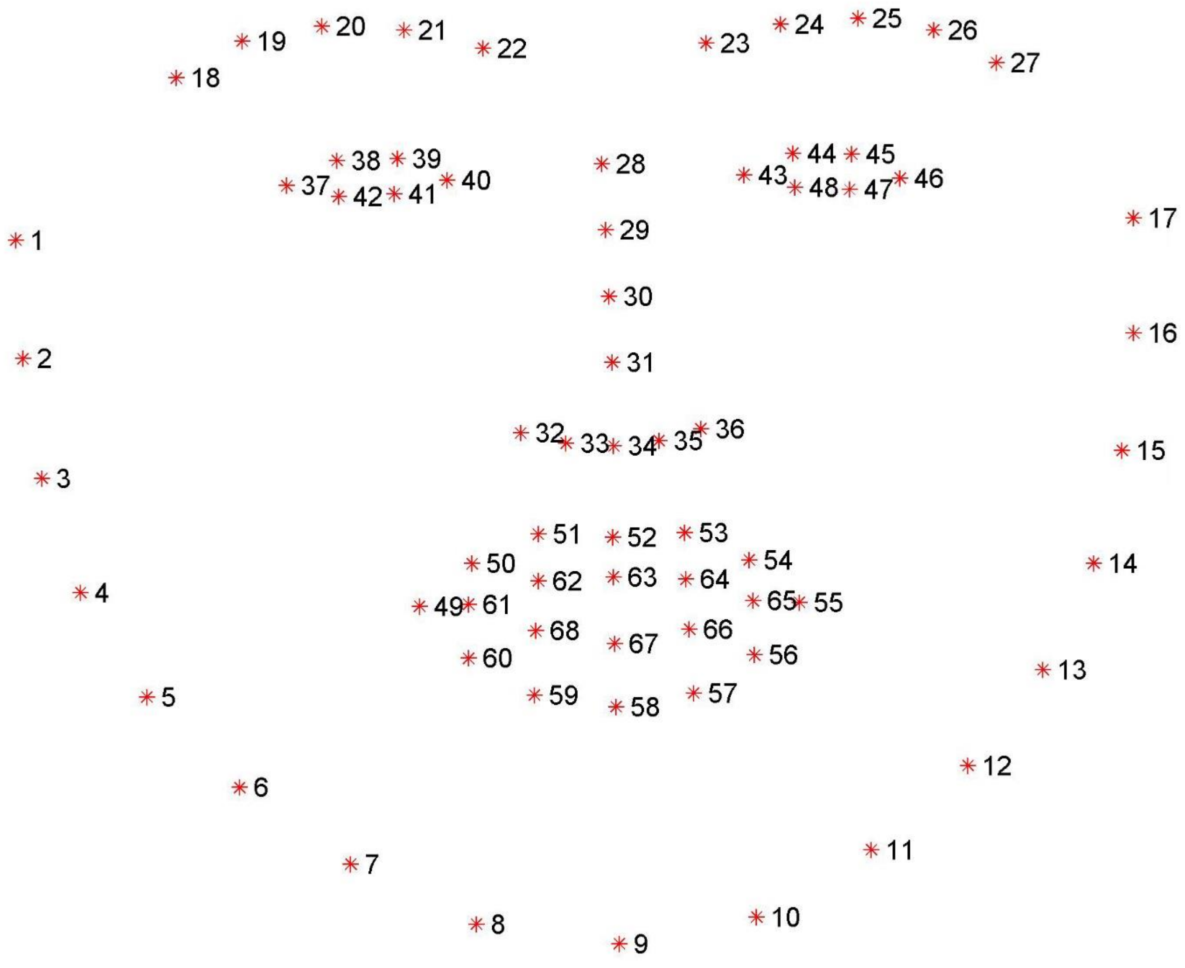
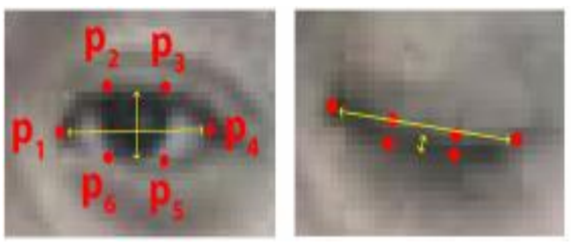


Figure 4.b.1: (Facial Landmarks)

For detecting blink, it is applied one mathematical method basing on distance. In figure 4.b.2 you can see open and closed eyes with land marks.[4]



Figüre 4.b.2 (Open and closed eye with Facial Landmarks)

Our mathematical method for detecting blink eye based on the calculation of below. First we calculate eye aspect ratio as in equation 4.b.1 for left and right eye.

$$EyeAspectRatio = (P2 - P6) \qquad (4.b.1)$$

After finding eye aspect ratio for left and right eye. The average of them is calculated as in equation 4.b.2

$$AverageEyeAspectRatio = \frac{EyeAspectRatioleft}{2} + \frac{EyeAspectRatioRight}{2} \qquad (4.b.2)$$

After finding AverageEyeAspectRatio, we threshold our system for one value and we detect whether our eye is open or close. After finding our threshold value, it is understood wheter the eye is open or not. You can see our Threshold in equation 4.b.3 and equation 4.b.4

$$AverageEyeAspect\ Ratio > Treshold\ value \qquad Eye\ Open \qquad (4.b.3)$$

$$AverageEyeAspect\ Ratio < Treshold\ value \qquad Eye\ Close \qquad (4.b.3)$$

The number of blinking is counted in ten second and then the system is supposed to output desired word in LCD monitor connected to Raspberry Pi.

To increasing speed of our code, the resolution is decreased.

5.) Budget

Product	Price	Number of Product	Link
Raspberry Pi 4 B	464,44 TL	1	https://www.direnc.net/raspberry-pi-4-model-b-
Raspberry Pi Camera V2	229.82 TL	1	https://www.robotistan.com/raspberry-pi-kamera-modulu-camera

3.5 Inch Lcd Screen (A) 320x480	154.13TL	1	https://www.direnc.net/raspberry-35inch-lcd-a-320480-waveshare?language=tr&h=eec9a565&gclid=CjwKCAjw26H3BRB2EiwAy32zhd5DxZ1lDYdC8oQturLj_c0rDGb_9fPj_OMDaJpX74BRsv3pYcdGkhoC9UsQAvD_BwE
Tpye C charger	29.90 TL	1	https://urun.n11.com/sarj-cihazi/samsung-type-c-orjinal-ithal-hizli-sarj-aleti-s8-s9-plus-note-10-P407614589
SanDisk Ultra 32GB	41,85 TL	1	https://www.hepsiburada.com/sandisk-ultra-32gb-80mb-s-microsdhc-microsdxc-uhs-i-hafiza-karti-sdsquns-032g-gn3mn-p-HBV0000084R79?magaza=Hepsiburada&wt_gl=cpc.6805.shop.nelk.foto-ssc&gclid=CjwKCAjwiMj2BRBFEiwAYfTbCiTcabXclTGDURqUSsSZBfg_Y8enZuq7WmXjjMPZxfzqX3I_ho6IgBoCtbYQAvD_BwE
SUM	920.14 TL	4	

6.) Revised Time Plan

Our project is completed therefore revised time plan is not prepared

5.) Referances :

1.) https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3

2.) 2013 IEEE International Symposium on Multimedia (https://ieeexplore.ieee.org/document/6746811?section=abstract)

3.) https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/

4.) https://www.raspberrypi.org/documentation/hardware/camera/

4.) https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

5.) http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf

6.) https://www.robotshop.com/en/35i-tft-lcd-320x480-touch-display-raspberry-pi.html

6.) Appendix

```
#!/usr/bin/env python3

# -*- coding: utf-8 -*-

"""

Created on Sun May  3 16:19:00 2020

@author: btknzn

"""

#You Have 22 selections, Just blink in 10 seconds and select the food you want to eat

#Our system counts your blink number from your first blick to 10 seconds later

import numpy as np

import cv2

import dlib

from math import hypot

import time

import tkinter as tk


def System():

    situation = True

    starttime=0

    eyeControlvalue=0

    lastvalueofeyecontrol=0

    counter=0

    while(True):
```

```
time.sleep(0.001)

ret, frame = cap.read()

# Our operations on the frame come here

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Display the resulting frame

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

    break

faces = detector(gray)

lastvalueofeyecontrol = eyeControlvalue

eyeControlvalue = eyeControl(gray, faces,frame)

#cv2.putText(frame,str(counter)+'blink number ',(0,400),font,1,(0,0,255),1)

print(str(counter)+"blink number")

#cv2.putText(frame,'Do you want '+systemTupel[counter]+'?',(50,50),font,1,(255,0,0),1)

if eyeControlvalue == 1:

    cv2.putText(frame,"Blinking",(50,150),font,1,(0,0,255),1)

    if situation:

        situation = False

        starttime=time.time()



#if eyeControlvalue == 0:

#    cv2.putText(frame,"not Blinking",(50,150),font,1,(0,0,255),1)
```

```
        #if eyeControlvalue == None:
         #   cv2.putText(frame,'no eye contact',(50,150),font,1,(0,0,255),1)

        print("To start system just blink. You have 10 second to make selection")

        #if situation:
        #    cv2.putText(frame,'To start system just blink. You Have 10 second to make
selecton:) ',(50,250),font,1,(0,0,255),1)


        #cv2.putText(frame,'To stop system press q ',(225,350),font,1,(0,0,255),1)


        if (eyeControlvalue==1) and (lastvalueofeyecontrol==0):
            counter= counter +1

        if ((starttime+10)<time.time()):
            if(situation==False):
                situation = True
                counter = 0


        #cv2.imshow('frame',frame)
        if (time.time()>starttime+10):
            situation = True



    cap.release()
    cv2.destroyAllWindows()



def changeSituation(situation,starttimesecond):
    timenow = time.time()
    starttimeplustensecond = starttimesecond+10
    if  timenow > starttimeplustensecond:
        situation = not situation





def eyeControl(gray, faces,frame):
    for face in faces:
        features = predictor(gray,face)

        R_Hori_left_x = features.part(36).x
        R_Hori_left_y = features.part(36).y
        R_Hori_left = (R_Hori_left_x,R_Hori_left_y)
        R_Hori_right_x = features.part(39).x
        R_Hori_right_y = features.part(39).y
        R_Hori_right = (R_Hori_right_x,R_Hori_right_y)
        cv2.line(frame,R_Hori_left,R_Hori_right,(0,255,0),1)
        R_upper_mid_x = int((features.part(37).x + features.part(38).x)/2)
        R_upper_mid_y = int((features.part(37).y + features.part(38).y)/2)
        R_upper_mid = (R_upper_mid_x,R_upper_mid_y)
        R_bottom_mid_x = int((features.part(41).x + features.part(40).x)/2)
        R_bottom_mid_y = int((features.part(41).y + features.part(40).y)/2)
        R_bottom_mid = (R_bottom_mid_x,R_bottom_mid_y)
        cv2.line(frame,R_upper_mid,R_bottom_mid,(0,255,0),1)
   #    find the lenght for both horizontal and vertical line
        R_hori_lenght = hypot(R_Hori_left_x - R_Hori_right_x, R_Hori_left_y -
R_Hori_right_y)
        R_ver_lenght = hypot(R_upper_mid[0] - R_bottom_mid[0], R_upper_mid[1] -
R_bottom_mid[1])
        features = predictor(gray,face)
        L_Hori_left_x = features.part(42).x
        L_Hori_left_y = features.part(42).y
        L_Hori_left = (L_Hori_left_x,L_Hori_left_y)
        L_Hori_right_x = features.part(45).x
        L_Hori_right_y = features.part(45).y
        L_Hori_right = (L_Hori_right_x,L_Hori_right_y)
        cv2.line(frame,L_Hori_left,L_Hori_right,(0,255,0),1)
        L_upper_mid_x = int((features.part(43).x + features.part(44).x)/2)
        L_upper_mid_y = int((features.part(43).y + features.part(44).y)/2)
```

```
    L_upper_mid = (L_upper_mid_x,L_upper_mid_y)

    L_bottom_mid_x = int((features.part(47).x + features.part(46).x)/2)

    L_bottom_mid_y = int((features.part(47).y + features.part(46).y)/2)

    L_bottom_mid = (L_bottom_mid_x,L_bottom_mid_y)

    cv2.line(frame,L_upper_mid,L_bottom_mid,(0,255,0),1)

    L_hori_lenght = hypot(L_Hori_left_x-L_Hori_right_x,L_Hori_left_y-L_Hori_right_y)

    L_ver_lenght = hypot(L_upper_mid[0]-L_bottom_mid[0],L_upper_mid[1]-
L_bottom_mid[1])

    L_ratio = L_hori_lenght/(L_ver_lenght+0.000000001)

    R_ratio = R_hori_lenght/(R_ver_lenght+0.000000001)

    ratio = (L_ratio + R_ratio)/2

    if ratio > 5:

        return 1

    return 0



cap = cv2.VideoCapture(0)

cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320)

cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)

print("başladı")

detector = dlib.get_frontal_face_detector()

predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

systemTupel =
("Pear","Quince","Apple","Plum","Cocount","Fig","Banana","Watermelon","Melon","Aprico
t","Cherry","Kiwi","Meat","Poultry","Fish","Seafood","Waffle","Ham","Bread","Creal","Ora
nge Juice" , "Muffin")
```

```
font = cv2.FONT_HERSHEY_COMPLEX

System()
```