

Joseph Boye & Kevin Tavara

ECE-350-001

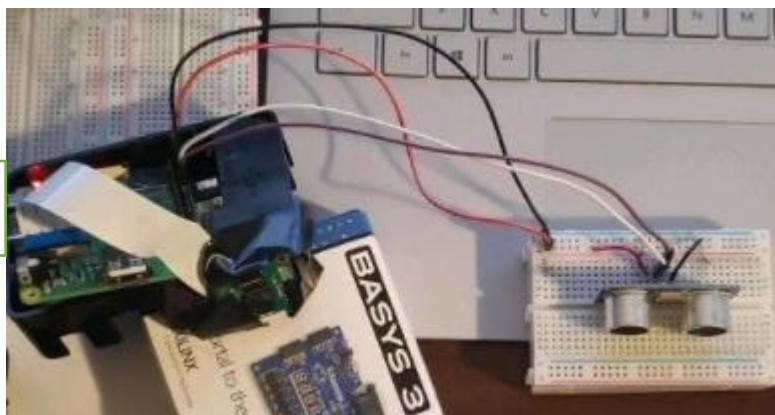
11.30.2020

Lab 7: Introduction to Computer Vision using OpenCV

Description: The purpose of this lab was to create an embedded system that can detect when a user has taken a seat using the HC-SR04 and capture a photo using the PiCam while recognizing the users face in the image.

Raspberry Pi Setup:

PiCam connected via ribbon cable to Pi



HC-SR04
Red: 5V
Black: Ground
White: GPIO23
Purple: GPIO 24

Captured Image:



Understanding Harr Cascade Classifier

Commented Code:

```
#!/usr/bin/python

# loads the necessary header files to enable the PiCamera, OpenCv, and their dependencies

import RPi.GPIO as GPIO

import time

from picamera import PiCamera

from time import sleep

import numpy as np

import cv2


# Adept Lesson 14 ultrasonic lab code

# Checkdist stores the distance between the chair from the HC-SR04 in feet


def checkdist():

    GPIO.output(16, GPIO.HIGH)

    time.sleep(0.000015)

    GPIO.output(16, GPIO.LOW)
```

```
while not GPIO.input(18):
```

```
    pass
```

```
t1 = time.time()
```

```
while GPIO.input(18):
```

```
    pass
```

```
t2 = time.time()
```

```
return ((t2-t1)*340/2) * 3.28 # converts meters into feet
```

```
GPIO.setmode(GPIO.BOARD) # sets GPIO to physical pin position
```

```
GPIO.setup(16,GPIO.OUT,initial=GPIO.LOW)
```

```
GPIO.setup(18,GPIO.IN)
```

```
time.sleep(2)
```

```
while True:
```

```
    distance_from_chair = checkdist() # stores the distance
```

```
    if distance_from_chair <= 1.5: #checks if the distance is less than one and half feet. This  
        indicates the seat is occupied
```

```
        camera = PiCamera()
```

```
        time.sleep(5)
```

```

camera.start_preview()

camera.capture('/home/pi/Documents/Lab7/image.jpg') # captures the image of the
person sitting in the chair

camera.close()

# Begins the facial recognition process and identifies the person's face sitting in the chair

# Uses the predefined classifiers that have already been trained for basic facial detection
(Haar Cascade)

face_cascade = cv2.CascadeClassifier('/usr/local/lib/python3.7/dist-
packages/cv2/data/haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('/usr/local/lib/python3.7/dist-
packages/cv2/data/haarcascade_eye.xml')

img = cv2.imread('/home/pi/Documents/Lab7/image.jpg')

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

faces = face_cascade.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces: # for loop places a blue rectangle around the face and a green
rectangle around the two eyes

    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2) # creates the rectangle
around the person's face

    roi_gray = gray[y:y+h, x:x+w]

    roi_color = img[y:y+h, x:x+w]

```

```

eyes = eye_cascade.detectMultiScale(roi_gray)

for (ex, ey, ew, eh) in eyes: # creates the green rectangle around the person's eyes

    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


cv2.imshow('img',img) # shows the image captured with the blue and green rectangles

cv2.waitKey(0)

cv2.destroyAllWindows()

# except KeyboardInterrupt:

GPIO.cleanup()

```

Harr Cascade Classifier: When it comes to detecting objects in an image using a computer, the Harr cascade classifier is widely used for object detection. It works by searching for specific features of the object in question and it moves a window over the image to search for detection. This is used for facial recognition because it can easily search an image for objects such as eyes or the features of a face.

References:

C. O. Manlises, J. M. Martinez, J. L. Belenzo, C. K. Perez and M. K. T. A. Postrero, "Real-time integrated CCTV using face and pedestrian detection image processing algorithm for automatic traffic light transitions," *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Cebu City, 2015, pp. 1-4, doi: 10.1109/HNICEM.2015.7393205.

S. Sander, "Object Detection using Harr-cascade Classifier," Institute of Computer Science, University of Tartu. Web. 2014. doi: <https://www.semanticscholar.org/paper/Object-detection-using-Haar-cascade-Classifer-Soo/0f1e866c3acb8a10f96b432e86f8a61be5eb6799#citing-papers>