HOME SURVEILLANCE SYSTEM

Winner Winner Chicken Dinner

COURSE: CS 121

Members: Aaron Zhang, Dongzhe Jiang

Date: May 4th, 2018

Table of contents

Introduction 3

Definitions, Acronyms, and Abbreviations 3

Project Details 4

Budget 5

Project Display 5

Time Spent 5

Final Reflection 6

Reference 6

Introduction

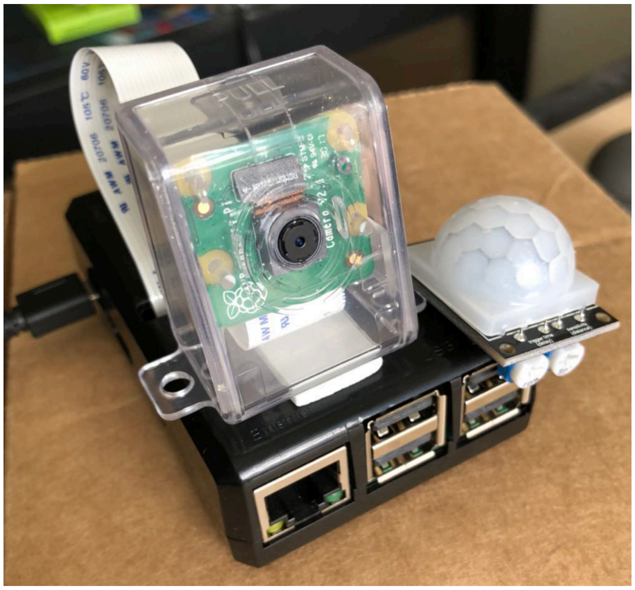
This is a home surveillance system with facial detection and recognition functions. Generally, this project consists of a raspberry pi, a camera and a motion sensor. It can be set to monitor your home. It will send warning message and captured image when it detects unfamiliar while the program is running.

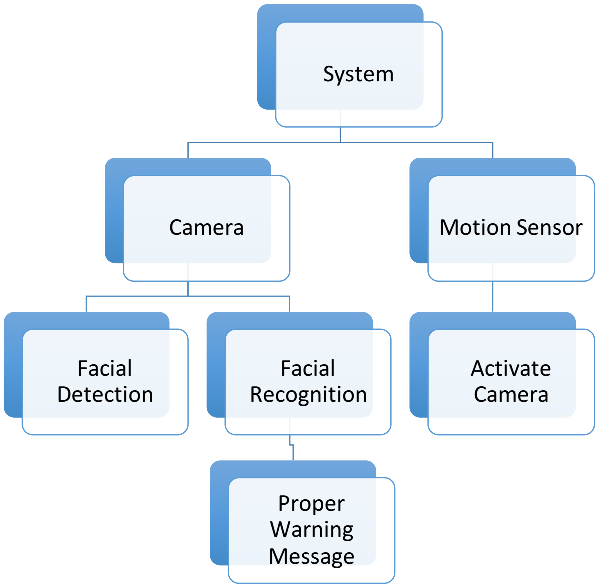
Definitions, Acronyms and Abbreviations

OpenCV: An open source computer vision and machine learning software library
cv2: OpenCV utilities package
cv2.VideoCapture(): Captures video
cv2.cvtColor(image, color): Converts color of image. We are using gray image.
cv2.imshow(): Shows everything the camera captures
cv2.CascaedClassifier('haarcascade_frontalface_default.xml'): Detects object
detectMultiScale(…): Initializes detection
LBPHFaceRecognizer_creater(): Create a face recognizer
cv2.FONT_HERSHEY_SIMPLEX: Create normal size sans-serif font
recognizer.predict(…): Return the confidence of recognizer

Project Detail

Motion sensor is used to detect motion. Once the sensor detects motions, it will activate the camera. Camera will detect and capture faces and activate recognizer. The recognizer will have a confidence towards the face that is being detected. If the confidence is less than a certain percentage, it will label it as unfamiliar. Since there is unfamiliar face, the camera will capture an image and name it with the date and local time, and will send a warning email to the user with the image that it captured.





Budget

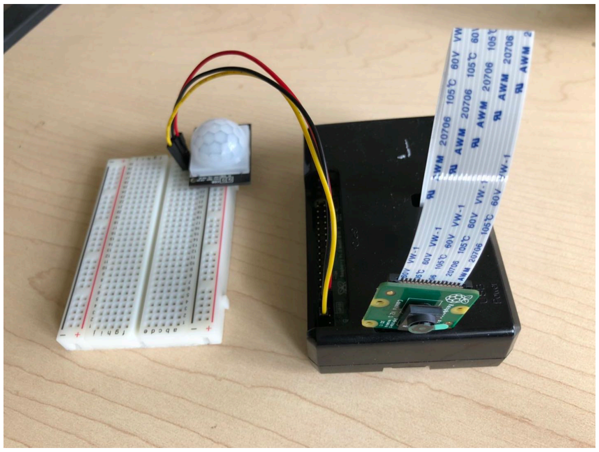
Expected Budget:

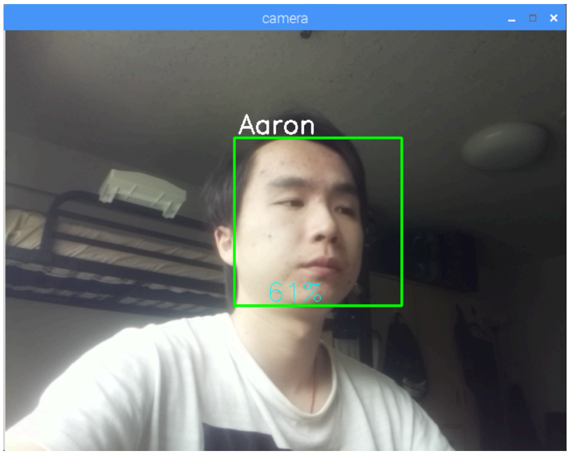
Raspberry pi kit $95
Camera $15
Motion sensor $10
Camera case $10
Total: $130

Actual Cost:

Raspberry pi kit $95
Camera $15
Motion sensor $10
Camera case $10 * 2 (one broken)
Total: $140

Project Display





In term of how to activate the program and how it works while detecting unfamiliar face, please see the following video: https://youtu.be/JfnJDKGWgwA

Time Spent

Estimated: 25 hours
Actual: 18 hours

Final Reflection

Overall, it is a fun project. We learned a lot about computer vision through this project, including the strategy like collection features and training program. Although we still don't understand how to do computer vision exactly, it motivated us to dig into this field. We faced lots of challenges at very beginning, such as how to do facial detection and recognition, and how to integrate everything like motion sensor and sending email. How to interact between OpenCV environment and GPIO was also another difficult. Fortunately, there are lots of resources that we can use. And we did figure out all of challenges. We also took away somethings. At first, we designed this program to recognize the object in front of camera is animal or human or other stuff in case that we cannot figure out how to do facial recognition. We skipped this step because we figured out this core section. Then everything else could go smoothly after that. Also, instructor and TAs help a lot on the interaction between program and user. They helped us to figure out how to connect to email server and send message to user.

References

OpenCV tutorial: https://docs.opencv.org/master/d9/df8/tutorial_root.html

Motion sensor tutorial: https://maker.pro/education/

Training code: https://github.com/Mjrovai/OpenCV-Face-Recognition/blob/master/FacialRecognition/02_face_training.py

Install opencv: https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/

OpenCV and GPIO tutorial: https://www.pyimagesearch.com/2016/05/02/accessing-rpi-gpio-and-gpio-zero-with-opencv-python/

All codes are submitted to the Code & Other Files section.