Bradley Allen

Talisa Powell

Professor Majumder

CSCIU509-01

25 April 2022

IoT Security Camera

Abstract:

For our IoT project we focus on a system that would be implemented for security purposes. The system alerts the owner of any unauthorized personnel that enters the owners space or home using a camera and facial recognition. Issues encountered were the accuracy of the facial recognition and low memory storage. Since the facial recognition used in our project compares pixels rather than learning, the accuracy of the facial detection is lowered. Our project uses a Raspberry Pi, which has low memory storage, this led to limited usage of the security system as a whole. The open-sourced library used was OpenCV, this is the backbone of our facial recognition. Our hardware consists of a magnetic relay that detects a threshold being accessed and turn on the camera to start the facial detection process. Once the process is complete, in respect to whichever scenario occurred, there may be an alert sent to the owner.

Ⅰ. Introduction

There are a multitude of reasons why one would want security and with the expansion of IoT there are plethora of ways to provide real-time security, and that’s what our project is about. Our project is ran on a Raspberry Pi and runs Python code for its processes. The facial recognition system, OpenCV, first uses the information it is given from a dataset of pictures to encode faces in order the store the information of that face. After the faces are stored, when a face is detected, it will compare based off of the stored information of the faces it already has to determine whether the face is authorized or not. There are four scenarios in our project that can occur. The first two scenarios do not send an alert to the owner: when there is an authorized person detected and when there is an authorized with an unauthorized person. The other two scenarios do send an alert: when there is an unauthorized person and when there are no faces detected when the camera is turned on. These notifications are sent via PushBullet and are displayed on the PushBullet app. The only issues that still exist are that since we are operating on a Raspberry Pi, the memory storage available is limiting and if one were to want a more accurate facial detection system, they would have to use a CNN (Convolutional Neural Network) which uses deep learning.

Ⅱ. System design

The hardware used consists of: Raspberry Pi, reed switch (magnetic relay), webcam, breadboard, 1k ohm resistor, cooling case for Raspberry Pi, keyboard, mouse, monitor, HDMI cable, and 3A power supply. First, an HDMI cable would be connected to a monitor and the Raspberry Pi along with a keyboard, mouse, and 3A power supply. Then, once the Raspberry Pi has been turned on and the code ran, the reed switch sends voltage when the circuit is closed (magnetic was moved). This action turns on the camera and begins scanning for faces.

The software used consists of Raspberry Pi’s operating system, Python, OpenCV, and the pi-face-recognition library. The Raspberry Pi’s operating system allows Python to be installed and ran. To encode faces, the dataset is used by OpenCV to create a .pickle file, which holds the data for the authorized faces. Detecting the faces is ran in a Python file that uses a database file (.xml) for facial landmarks to compare and the .pickle file to compare to. Inside the same file, PushBullet is initialized and can send a notification to the owner.

Ⅲ. Results Analysis

The result of the project is a video stream capable of recognizing faces, determining whether the scenario should notify the owner, and being handled automatically. Problems encountered along the way included difficulty downloading the libraries required, overheating of the hardware, low voltage power supply, low frame rate, and low memory. To fix the overheating and low voltage, a cooling case was used for the Raspberry Pi along with a stronger power supply. To combat the low frame rate, the video stream window was made smaller so the comparisons needed to recognize a face were reduced. The only outstanding issue is the memory storage for a Raspberry Pi is extremely low for this project’s purpose.

Ⅳ. Related Works

While IoT facial recognition cameras are widely commercialized, this project’s implementation is similar to Suraj Pawar’s publication of “Smart Home Security Using IoT and Face Recognition.” In Pawar’s publication, a passive infrared and ultrasonic sensor are used in combination with the camera. The infrared sensor is used to detect movement, replacing our project’s reed switch, and the ultrasonic sensor detects if the door has been broken into. With Pawar’s implementation, the facial detection allows the door to be unlocked or sends a notification to the owner if there is an unidentified face.

Ⅴ. Conclusions

This security system can be implemented in homes of those who are not regularly home, whether it be for work or pleasure. If this project were to be re-created, a computer with a dedicated GPU would need to be used to have the processing power to run real-time facial detection.

VI. References

S. Pawar, V. Kithani, S. Ahuja and S. Sahu, "Smart Home Security Using IoT and Face Recognition," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697695.

“OpenCV Library” OpenCV, GitHub, https://github.com/opencv/opencv/tree/4.5.5.

Table of Time Contributions

|  |  |  |
| --- | --- | --- |
| Week | Bradley | Talisa |
| 1 | 1 hour | 1 hour |
| 2 | 4 hours | 4 hours |
| 3 | 5 hours | 5 hours |
| 4 | 4 hours | 4 hours |
| 5 | 3 hours | 3 hours |

Link to Project Demo

[https://emailuscupstateedu-my.sharepoint.com/:v:/g/personal/powelltk\_email\_uscupstate\_edu/EVfO4eqsgBJGkx\_CwCLHlg8BbjiQeVuwwWrovBlO9xTmdA?e=Eaidxk](https://emailuscupstateedu-my.sharepoint.com/:v:/g/personal/powelltk_email_uscupstate_edu/EVfO4eqsgBJGkx_CwCLHlg8BbjiQeVuwwWrovBlO9xTmdA?e=Eaidxk%20)

Pictures of Project

 

 

 

 





  







