Corbi Home

CS 480I/580I Project

Ethan Elliott, Joseph Sanchez, Siddharth Singh
Department of Computer Science, SUNY Binghamton, Binghamton, NY, United States
eelliot4@binghamton.edu, jsanch49@binghamton.edu, ssing117@binghamton.edu

1 Introduction

Wireless home security provides the same level of threat protection as a tradition system with the convenience of wireless setup. Current systems may achieve similar levels of protection, but with the hassle of setting up a hardwired system. Corbi Home aims to provide a home security solution that meets or exceeds current standards while providing a simple, wireless setup. Using facial recognition cameras and motion sensors, Corbi Home ensures the homeowner is always aware of where a threat may lie near or inside their home.

2 Implementation

In this section, we will introduce the tools we utilized and the steps we took to implement the home security system.

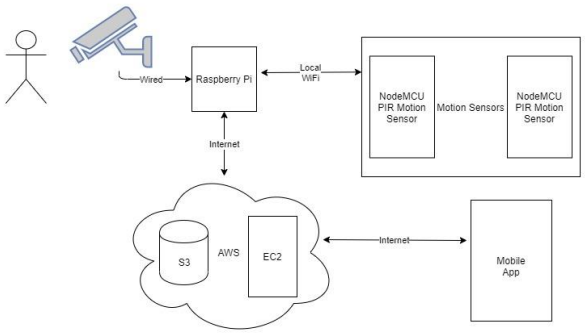

Figure 1: Final design diagram for the complete system

2.1 Hardware

To achieve a facial recognition camera, we utilized a Raspberry Pi 4 with an attached Raspberry Pi no-IR camera. This unit provided fast and simple facial recognition locally. For our motion sensor network, we utilized NodeMCU boards, each attached to a PIR motion sensor. Each board could communicate directly to the master server.

2.2 Data Collection

The home security system begins with the facial recognition camera running on the Raspberry Pi. Ideally this camera would be mounted outside a homeowner's exterior door, monitoring for people entering the home. However, for our demonstration purposes we simply pointed the camera at known and unknown faces. The Raspberry Pi constantly polls the camera for images and scans each image for a face. If a face detected, the image is compared against a whitelist of known faces. If an unknown face is recognized, the Raspberry Pi sends the image to a remote web server to be stored and displayed at the request of the homeowner.

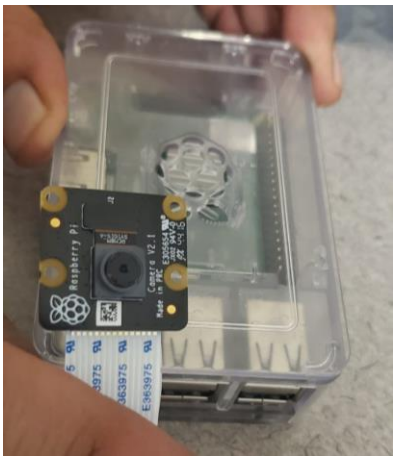

Figure 2: Raspberry Pi 4 and camera module

The next piece of the home security system is a collection of motion sensors, each attached to a NodeMCU. Each motion sensor is assigned to a room and when motion is detected, report back to the remote web server which room motion has been detected in using Mosquitto. For our demonstration purposes, we set up two NodeMCUs, each with a motion sensor attached, and pointed them in opposite directions. We simulated walking through a room by waving our hands in front of one of the two motion sensors.

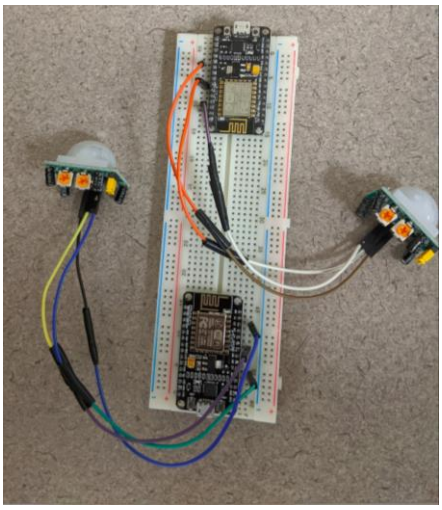

Figure 3: PIR motion sensors attached to NodeMCU boards

On the remote web server, a logger script listened for any data published by a motion sensor unit. When a message is published, the script logs the message to a single file that will be polled by the Android application.

2.3 End-User Interaction

To provide a simple and useful way for an end-user to interact with the home security system, we built an Android application. The application consists of a three-tabbed design: welcome tab, intruder photos tab, and list of rooms with motion detected tab. The application makes an HTTP request for the list of intruder photos and the list of rooms motion has been detected in from the remote web server which sends the data through an HTTP response. Each photo is rendered by their URL assigned by the web server.

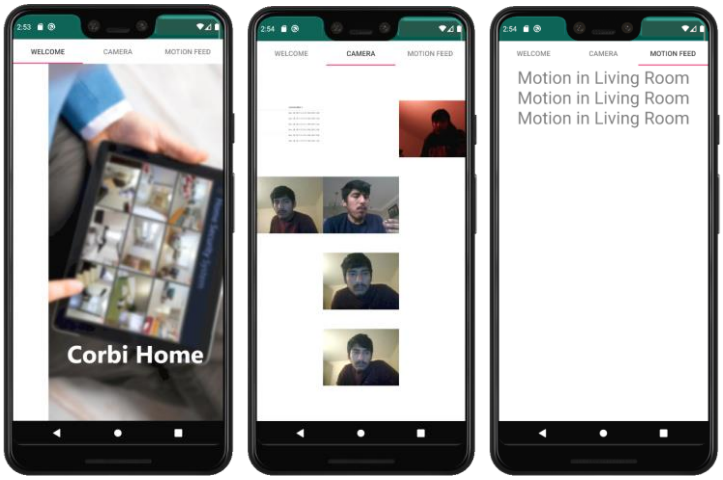

Figure 4: Screenshots of the Android application.

3 Evaluation

To evaluate the project, we started everything up together and let it run on various inputs while monitoring the outputs for correctness.

3.1 Face detection and recognition

To evaluate the face detection and recognition feature on the Raspberry Pi, we loaded one of the team member's faces to the whitelist of known faces. Using the camera, we presented each of our faces to the camera. We found that the detection algorithm was nearly perfect at finding a face. The recognition algorithm was nearly as good as detection, but still worked very smoothly. We found that false negatives were more common than false positives.

3.2 Motion sensors

To evaluate the motion sensors, we connected each NodeMCU to the remote web server and triggered the motion sensor for each. We then monitored the remote web server for incoming messages from the NodeMCUs. We found that the motion sensors were relatively inconsistent. However, when the motion sensors were working, the NodeMCUs worked flawlessly each time.

3.3 Android application

The final piece of the project was an Android application used to present the gathered information to the end-user. After confirming on the remote web server that the camera and motion sensors were indeed gathering the correct information, we could evaluate the Android application by comparing the information it should gather from the web server with the information it gathered. The Android application worked perfectly with the web server and always connected without issue.

4 Conclusion

Corbi Home was designed to provide homeowners with the same level of home security you would get from a traditional hard-wired system but with the convenience of wireless technology. By connecting all your components wirelessly, the system is compatible with any home with an existing wireless network. We found that the Raspberry Pi and camera worked well enough to demonstrate for facial detection and recognition but would ideally be better for a real-world application. The motion sensors were unreliable and would need to be improved, but the wireless

boards used to connect each sensor to the remote web server (NodeMCUs) worked perfectly. Finally, the Android application we developed to display all the gathered information to the end-user seamlessly connected to remote server and always displayed the desired information.