A Facial Recognition Approach to IoT Home Solutions

Group 12 | ECE 499 | Summer 2021 | Supervised by Dr. Alexandra Branzan Albu

Grayson Halliday | Luke Evans | Peter Ogilvie | Thomas Appaix | Andrew Derocher

University of Victoria

Project Background

- Machine learning and Internet of Things are rapidly growing industries
- These advancements are being incorporated into home automation applications allowing the user to customize control of their home
- Open-source platforms like TensorFlow and OpenCV are making machine learning more accessible for a variety of applications

Problem Statement

- Current home automation systems determine when a user is home by detecting whether their personal devices are connected to WiFi.
  - Residents that do not carry a mobile device are excluded from these platforms.
- Current facial detection systems target the home security market, and only allow for notifications rather than automations.
- There exists an opening in the market for a facial recognition system compatible with third-party home automation platforms.

Objectives

Develop a facial recognition system for home automation, with the following features:

- A detector module capable of
  - Detecting motion
  - Taking photos
  - Predicting the identity of the resident seen
  - Sending results
  - Perform the above in less than 10 seconds with at least 95% accuracy
- A custom power supply and battery management system able to power the detector module
- Bill of materials should be under $150, for competitive pricing in the home automation market
- Home Assistant integration with existing IoT devices

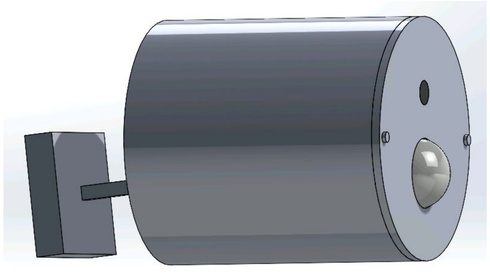

Fig. 1. Detector module case design

Design Methodology

Firmware

- Developed in Python on the Raspberry Pi Zero W
- Remains idle between motion detector triggers
- Uses OpenCV's Cascade Classifier to determine if faces are present and crops to a 500x500 image

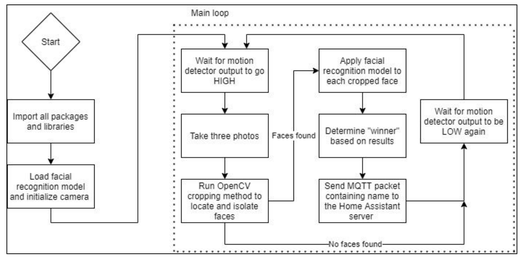

Fig. 2. Firmware flowchart

Facial Recognition Model

- Input
  - 500x500 pixel image
- Output
  - Array of classification probabilities
- 3-Layer convolutional neural network
- Trained on images of the 5 group members
- Uses TensorFlow Lite model architecture due to Raspberry Pi Zero W CPU constraints
  - Full TensorFlow model is possible if Raspberry Pi 4 is used

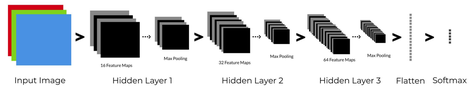

Fig. 3. Convolutional neural network architecture

Home Assistant

- Free open-source home automation software
- Allows custom automations, triggered by connected devices
- Module sends identity via MQTT
- WiZ RGB smart light bulb used for in-person demonstration

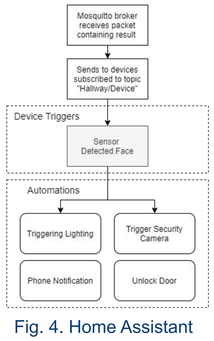

Fig. 4. Home Assistant flowchart

Power Supply

- Linear Power Supply Requirements:
  - 5.1V, 2.5A output
  - Supports Raspberry Pi Zero W and peripheral loads
- Simulation Results:
  - 0.11 mV output ripple at full load

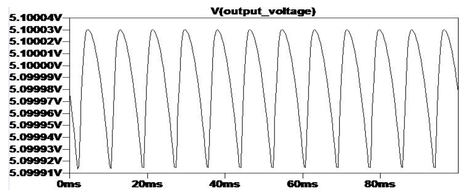

Fig. 5. Power supply simulation results

Battery Management System

- Expected runtime of 15 hours on a single charge
- 5.1V, 2.5A output
- 65x75 mm PCB for mounting a Raspberry Pi Zero

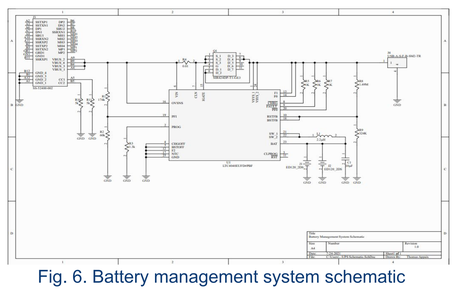

Fig. 6. Battery management system schematic

Final Design

- Prototype system successfully meets most objectives
- Raspberry Pi Zero W takes 60-80 seconds to complete a facial recognition sequence
  - Using the more powerful Raspberry Pi 4 solves this issue and meets the 10 second runtime objective
- Case designed to contain detector module
- Total bill of materials is $114.78 (excluding case)

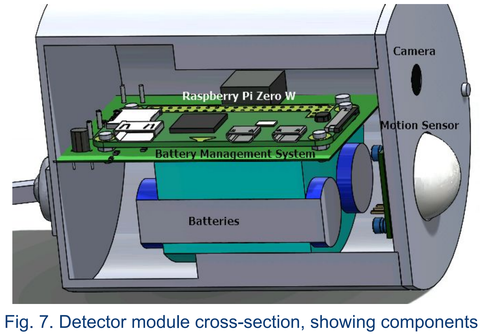

Fig. 7. Detector module cross-section, showing components

Results and Validation

- Group members each triggered 10 detection sequences
- 98% recognition accuracy, 95.2% mean confidence value
- Home Assistant was able to turn a smart RGB light bulb different colours depending on the identity found
- Power supply passed test simulations

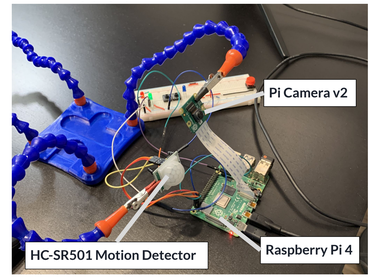

Fig. 8. Test setup with Raspberry Pi 4

Conclusion and Recommendations

- Prototype successfully integrated facial recognition within a home automation platform
- Recommendations for future work:
  - Replace the Raspberry Pi Zero W with a more powerful/specialized microcontroller
  - Validate CNN architecture on demographically-diverse datasets
  - Develop intuitive model training method, similar to Apple Face-ID
  - Configure device auto-discovery via MQTT

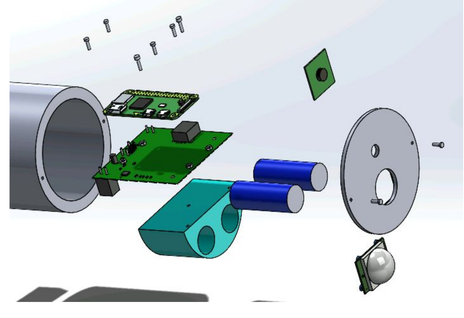

Fig. 9. Exploded view of the detector module