unlv_logo **University of Nevada Las Vegas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Class:** | EE498 Senior Design II | | | **Semester:** | Spring 2020 |
|  | | | | | |
| **Group members:** |  | **Project topic:** | *Keyless-Entry Door Using Facial Recognition* | | |
| *Adrian Ruiz*  *Bryan Takemoto* |  | | | |
| **Document:** | Progress report I | | |

**Abstract**

The Keyless-Entry Door (KED) using facial recognition is developed to provide security and convenience to the front door. Using a camera, the KED captures images of the person’s face and verifies the identity using a convolutional neural network. Once the system verified the person’s identity, the KED unlocks the door using a motor. This gives the impression of a seamless entry as if the door was not locked. If the KED concludes that the identity of the person is invalid, the door remains locked. Inspired by *smart homes*, the KED aims to be integrated into this system. For instance, if a *smart smoke alarm* detects a fire in the house, the *smart smoke alarm* will automatically signal the KED to unlock the door for firefighters, thus, saving the firefighter the trouble of breaking down the door. Unlike traditional locked doors, the KED has sensors to detect unusual amount of forces applied to the door. If the force is large enough, the KED alerts the authorities of a possible break-in. Although facial recognition is the primary method to unlock the door, the KED offers keypad and manual entry with key. An application of the keypad is that couriers can use a temporary key code to open the door and safely drop off valuable packages inside the house. As more embedded systems are integrated into daily lives, the KED pursues to be part of this future market. Hopefully, the KED will see itself as an early adopter to *smart homes*.

**Current state of the project**

Write the synthetic description of the current state of your project. 2-3 paragraphs.

Currently, we have the ATmega communicating with the Raspberry Pi, and we have facial recognition running on the Pi as well. The ATmega328 works as a secondary microcontroller to monitor the accelerometer, read inputs from the keypad, and control the stepper motor. For facial recognition, we use a pre-trained CNN. We train the CNN on a separate device and transfer the model to the Pi for use. No actual training is done on the Raspberry Pi, currently. When the camera attached to the Pi recognizes a face, it sends the frame from the video to the CNN, and the CNN returns a confidence value. If the confidence value is high enough (i.e. how confident the CNN recognizes a person), a recognition is made.

With regards to the CNN, there is a possibility of a false positive, and it runs slow on the pi. We fix the false positive problem by testing multiple frames. If a set number of frames return positive, then the recognition is a success. In terms of speed, sending a frame through the CNN also takes about 30 seconds to return a value. We have a couple solutions for this problem. We believe loading running Raspbian on a SSD will increase the speed of execution.

**Current state and roadmap of the project**

Fill the table. Provide the state that your project was at the beginning of the semester. Due dates must comply with the due dates provided in the WebCampus (see the WebCampus for the most recent version of the calendar). Add more rows if necessary.

|  |  |  |  |
| --- | --- | --- | --- |
| **Completed steps** | **Step completed** | **Date completed** | **Comment** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **Incomplete steps / planned goals** | **Step/goal** | **Date to be completed** | **Comment** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Current problems**

Describe current problems you have. Categorize each problem as: easy/moderate/high severity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem** | **Severity** | **Idea(s) how to solve** | **Comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**All roles in the project**

In the table below, briefly describe roles and skills required to do the project

|  |  |
| --- | --- |
|  | **Required skills** |
| **Role 1** |  |
| **Role 2** |  |
| **Microcontroller programmer** | * Knowledge of ATMEGA168 microcontroller * C++ programming * AVR Studio experience |
| **Image processing specialist** | * Knowledge of image processing algorithms |
| **CAD designer** | * CAD skills sufficient to design the device case |

Table 3. Roles & skills

**Team member roles**

Describe who is doing what role in the project

|  |  |
| --- | --- |
|  | **Roles assigned** |
| **Team member 1** |  |
| **Team member 2** |  |

**Members’ tasks**

Create a column for each group member.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Contribution of the member to each task 🡻** | | |  |
| **Member name 1** | **Member name 2** | **Member name 3** | **Task/step completed? (yes/no)** |
| **Task/step 🡻** |
|  |  |  |  |  |
|  |  |  |  |  |

**Current Timeline**

|  |  |
| --- | --- |
| **Week** | **Actions planned** |
| #6 (Oct.2-Oct.8) | * 1 * 2 * 3 * 4 * … |
| #7 (Oct.9-Oct.15) | * 1 * 2 * 3 * 4   … |
| #8 (Oct.16-Oct.22) | * 1 * 2 * 3 * 4   … |
| #9 (Oct.23-Oct.29) | * 1 * 2 * 3 * 4   … |
| #10 (Oct.30-Nov.5) | * 1 * 2 * 3 * 4   … |
| #11 (Nov.6- Nov.12) | * 1 * 2 * 3 * 4   … |
| #12 (Nov.13- Nov.19) | * 1 * 2 * 3 * 4   … |
| #13 (Nov.20- Nov.26) | * 1 * 2 * 3 * 4   … |
| #14 (Nov.27- Dec.3) | * 1 * 2 * 3 * 4   … |

**Comments**

Add any comments related to the project

Remove all text in green.