

**University of Nevada Las Vegas**

**Department of Electrical and Computer Engineering**

**EE498 Senior Design**

Spring 2020

**Project Title**

Final Project Report

**Insert project photo here inside of this box**

**Group Members:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Name (Print) |  | CPE/EE/ME |  |
|  |  |  |  |
| Name (Print) |  | CPE/EE/ME |  |

Instructor: Dr. Grzegorz Chmaj

Faculty advisor:

**Table of contents**

All sections and numbered paragraphs must be listed here along with page numbers

**Abstract**

Keyless Entry Door (KED) is developed to provide convenience and security to the front door. The device is inspired by the future of smart homes. KED offers convenience by unlocking the door using FaceNet, a CNN for facial recognition. FaceNet analyzes the person’s facial structure and verifies their identity. If the person’s identity is valid, the motor will unlock the door, otherwise, the door remains locked. Thus, giving a seamless entry without the need of a key. Security is also improved by implementing a force detecting sensor onto the device. If an unusual amount of force is applied to the door, KED will alert the owner of a possible breakin or dispatch authorities if necessary. Further functions of KED include the ability for couriers to safely drop off packages, generate digital keys for friends or family, and access the camera at the front door.

**Introduction & background**

Keyless Entry Door (KED) identifies people’s faces in a video stream, compares them to images in a database, and recognizes known faces. If an individual’s face is recognized, the door unlocks itself automatically. Additional features include a key generator which allows entry through a keypad, and an accelerometer to detect a break in. KED offers convenience as the user does not have to pull out his or her key in order to open the door. More than one-third of people have been a victim of package theft. We wish for KED to integrated with courier services by allowing the courier to open the door and place the package inside.

The number of smart home products has been increasing with the market expecting to increase by 1.2 billion by 2022.

Facial Recognition has been implemented in a variety of products and services. One example is the service FacePRO from Panasonic. FacePRO’s matches a person’s face using live video to a database of registered faces and alerts the user of matching faces. It includes up to 20 cameras on a server and up to 30,000 known faces. A similar product is FaceVACS from Cognitec. FaceVACS also identifies peoples’ faces from video streams and recognize known faces. FaceVACs also provides services like frequent visitors, generate demographic statistics, and more depending on the version of the product bought. Other products that closely resemble our project are smart doorbell products. Standard features of these products are two-way communication, video recording, and motion sensors. Google Nest has the features mentioned above, along with a facial recognition system. The facial recognition is only used to alert the user if a face is recognized.

Our project is different from the above-mentioned products. Products like FacePRO and FaceVACS recognize people and alert the user, they are not products the general consumer will buy. They are more focused on business and public use. Recently, states have been coming together to ban facial recognition in public settings, so these products might not be available for businesses in most states.

The products above do not mention how they perform facial recognition, but we assume they use deep learning, like our project, in order to the accuracy they mention. Deep learning is a class of machine learning in which neural networks are used to extract information from input. A convolutional neural network (CNN) consists of an input layer and an output layer, along with multiple hidden layers. For our project, an image of a face is sent into a CNN, and its features are extracted as it goes through each of layer of the CNN. It returns a probability that the face in the image is a certain person. There are also different CNN models, and each are designed for a specific input. We chose the FaceNet model, since it specializes in facial recognition. When an image is passed in, FaceNet calculates how close that the image is to an anchor image. We use the probability calculated to determine whether to unlock the door. The user can also use the keypad to unlock the door. The passcodes are randomly generated and expire in order to increase the security.

**Current Market Solutions**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Vendor** | **Video Resolution** | **Misc.** | **Facial Recognition** | **Function 2** | **Price** | **Comment** |
| **FacePro** | PanaSonic | NA | Up to 20 Cameras per Server | In-House algorithm  Above 90 percent accuracy |  | TBD by User | Needs to be ordered and installed by Vendor |
| **Google Nest**  **Hello** | Google | 1080p | Enables two talk communication | Recognizes faces and sends alerts to user |  |  | Needs to be ordered and manufactured by vendor |
| **KED** | You | 1080p | Email Alerts of break in | FaceNet  Highly accurate |  |  |  |

Table 1. Comparison of available devices

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Resolution** | **Battery life** | **Price** | **…** | **Strengths** | **Weaknesses** |
| **FacePro** | NA | Long | Very high |  | * High Amount of Security * Massive amount of Saved Data | * High price * Public Installation is illegal in some states |
| **Nest Hello** | HD | Long | Affordable  ($229) |  | * Long battery life * Affordable price | * Low resolution |
| **KED** | HD | NA | Affordable  ($240) |  | * High resolution * Low price |  |

Table 2. Strengths and weaknesses of available devices

**Research results**

The project required a computer in order to run the facial recognition program. The program uses TensorFlow’s python package, so a microcontroller would not be able run the program. The RPi was chosen since it is a microcomputer which can run python programs. It has a four core CPU which allows us to thread our program. It is also compact. Due to the lack of GPIO pins on the raspberry pi, we needed something to controller the motor, read in accelerometer data, and read in keypad data. The Atmega328p was chosen as many people are familiar with this microcontroller. For our camera, KED was initially tested the facial with a Logitech 720p webcam. KED later switched to a raspberry camera which is 1080p. With a higher quality camera, the facial recognition program was more accurate.

For facial recognition, there are several available options. Initially, KED used the Haar Cascades to do face recognition and face verification. Haar uses pixel sums to identify features in a face. Haar requires multiple images to describe a face with accuracy because it is a weak classifier, its accuracy is slightly better than random. Due to its somewhat poor verification, Haar cascades are only used in detecting a face. OpenFace is a deep learning facial recognition model that was considered for KED. OpenFace is a neural network that takes in a face image and transforms it into a specialized embedding. It does not use a Euclidian space compared to other neural networks. Because of its embedding, it is one of the faster neural networks available, but some accuracy is lost because of this. Facebook’s VGG neural network was an available option for facial recognition. While it was accurate, the neural network was very CPU intensive on the RPi. Facial recognition ran extremely slow on RPi.

FaceNet was the deciding neural network. FaceNet was selected for KED because it accomplishes both accuracy and efficiency. FaceNet achieves more accurate results when compared to OpenFace and does not slow the verification process like the VGG. FaceNet is unique in fact that it is two neural networks connected. The first neural network in FaceNet maps the face into a 3D Euclidean space. It also optimizes the Euclidean space for the next neural network. The second neural network determined which convolutions to use on the Euclidean space. A convolution is a way filter information passed into a smaller size. The output of second neural network is compared to the expected face. This comparison is done by computing the distance between the two Euclidean spaces. If the two faces are identical, then the distance is short, else the two faces are unrelated.

**Specification of the project**

**Functionality & conceptual design:**

One thread builds the CNN for facial recognition, and it takes around thirty seconds for the CNN to be fully built. Once the CNN is fully built, the main menu is displayed. Along with the main menu, a window opens that displays the camera feed. In the background, KED is measuring accelerometer data, and a facial recognition system. From the main menu, the user can unlock the door manually, change settings, and manage keys. KED is also constantly scanning for a face. If KED detects a face, several actions happen. On the camera feed, a rectangle is displayed around the face detected. In the background, the face detected is sent to through the CNN. If the face detected is a recognized user, the door is unlocked. Also, the recognized user’s face is displayed in the camera feed. A typical use case can be seen in figure ()

A screenshot of a cell phone

Description automatically generated

As mentioned above KED’s main program uses threads. The number of threads and their functions can be seen in figure (n). The first thread controls the main menu display. It uses a python package used Cursors in order to cleanly display options and information. From the main menu, the user can toggle different options. If the users selects the “Disarm/Arm security” option, the security is armed or disarmed in response. “View Settings” allows the user to change sensitivity of the accelerometer and the distance needed for verification. For accelerometer, the more sensitive it is, the threshold to trigger an alert is lowered. For the distance, the less sensitive it, the threshold needed for verification is lowered. The user can also manually lock and unlock the door from the main menu. The user can see pervious keys used with the “Key History” option. With the “Generate Key” option, the user can make different keys that can unlock the door using the keypad. The user has the options of setting the expiration date of these keys. The user can also determine if this key is for a courier or for a friend. The user can also remove keys with the “Remove Key” option.

A picture containing computer, meter, display

Description automatically generated

* Use diagrams and descriptions to describe **how** the device is working and **what** it is actually doing (use flow/case diagrams)
* Include use-cases:
  + You have actor1 doing action A, the system responds with the response A'
  + You have actor2 doing action B, the system responds with the response B'
  + (example) actor=user walks into the kitchen and says "coffee". System starts the coffee-maker and pours the coffee in the mug.
* Block diagrams must be here
* Modular description must be here

**Architecture:**

KED is two different modules, the RPi and the Atmega. The RPi manages the Key Database, the Random Key Generator, the Security Settings, Camera, and the Facial Recognition. The Atmega communicates with the accelerometer, and the motor. The RPi and the Atmega both communicate to each other via UART.

The RPI runs the main program which controls and manages components connected to it. The components it manages mostly act independently.

The Key Generator generates random keys that act as passcodes to the door. The time and date the key was created is also saved and assigned to the respective keys.

The Key Database manages the keys created. The user can view the keys created and delete keys manually. Keys are automatically deleted after a certain amount has passed since the key’s creation.

The Camera sends a live feed to the RPi which is displayed on screen. It is also where face detection occurs.

The Facial Recognition is activated when the camera detects a face. This component takes a frame from the camera when a face is detected. This frame sent through the CNN to determine if it is a recognized face.

The Alarm module alerts the user when KED determines a break in. It will send an email to the user, call the user, and play an alarm sound. It determines a break in from the accelerometer data.

The Atmega controls the motor and accelerometer, and it communicates with RPi via UART.

* Focus on architecture details in this section: describe each object and all signals
* Start with architecture overview. Provide high-level diagram, describe each component briefly, including: input(s), operation, output(s)
* Start the paragraph for each component, describe it and then expand the component into lower level, providing the diagram for that too. Repeat expand until you reach desired details level.
* Conclude with the diagram containing all the objects expressed at low level.
* All architecture details should be described in this subsection: objects, their inputs/outputs, relation to other objects.

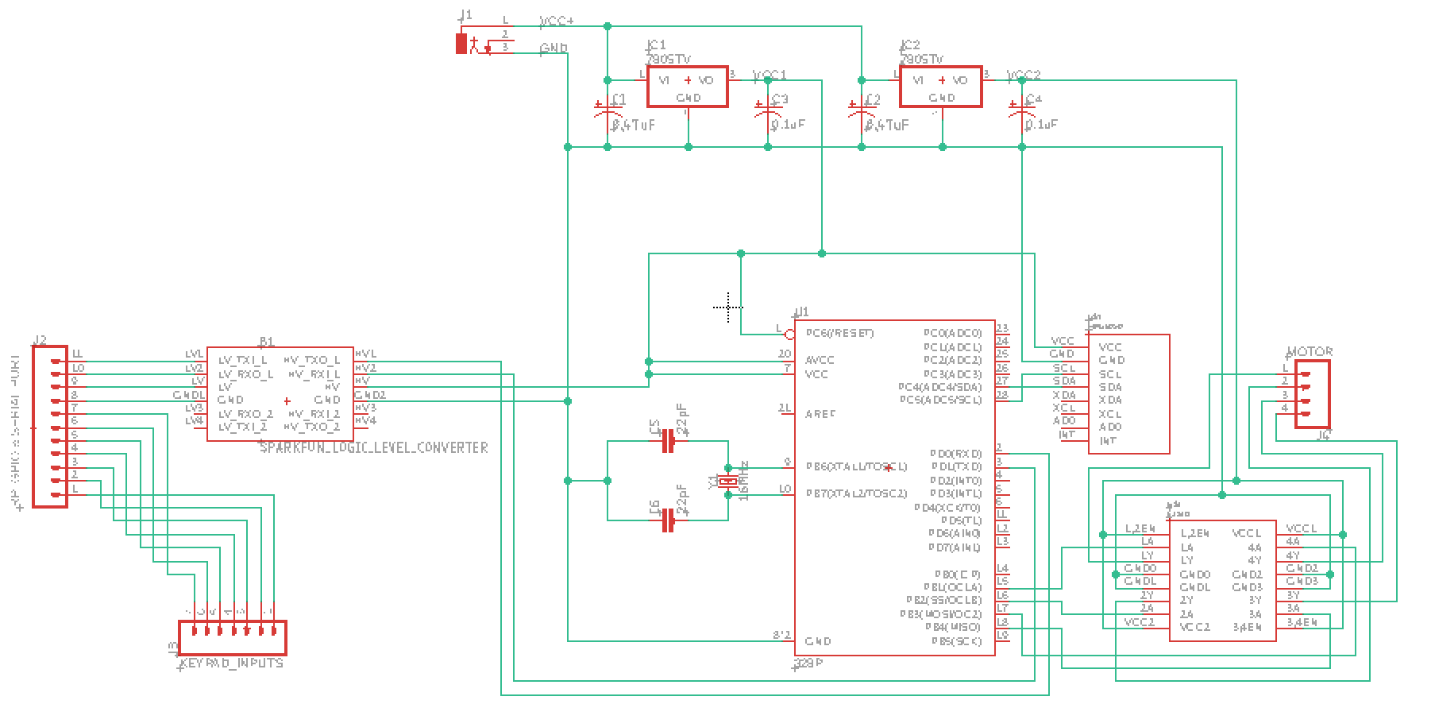
**Design**

The Key Generator generates passcodes at the length of 5 digits. The time and date the key was created is also saved to the key. The date and time are found by using a python package called datetime. An expiration date is also given to the key upon creation. The user can determine on which day the key will expire. The user can also determine who will be using this key such as friends or couriers. Keys created are saved into a key list file. The expired key checker will constantly go through this key list to find any expired keys. Any expired keys are deleted off the key list.

The camera is attached to the RPi via (USB or RPi Camera). In order to display the camera’s live feed, the python package, OpenCV, is used. Using OpenCV, the main program opens a window for the camera feed. Along with live feed, OpenCV is used for face detection, but not used for facial recognition. It uses Haar Casscades in order to determine if a face is detected. A rectangle is drawn around peoples’ faces on the camera feed to indicate that a face has been detected.

The facial recognition starts when a face was detected. When a face is detected, the frame is used for facial recognition. The frame is cropped and resized, so that frame is only the detected face. The detected face is sent through the CNN and formatted for image comparison. The formatted image is compared to an image of a registered user using cosine similarity. If the similarity is underneath a specific threshold, the face detected is considered recognized, and the door is unlocked. Also, the recognized person’s name is displayed on the camera live feed.

The alarm module is activated when a break-in is detected. A determined if the accelerometer’s values pass a threshold set by the user. When the break-in happens, the frame with the culprit’s face is saved. This frame is than emailed to the user along with a message. The email is sent using python package email. The user’s phone is also called in order to alert them. The phone call is triggered by using a Webhook from IFTTT. It uses a GET request to trigger the phone call. After the phone call and email, KED will start playing an alarm. The simpleaudio python package is used in order to play sounds. KED will keep playing the alarm until it is deactivated.

The Atmega is used to communicate with the motor and the accelerator. The type of motor KED uses is a stepper motor which are controlled by using discrete phases. The Atmega reads in accelerometer data using I2C. The Atmega also takes in inputs from the keypad. Both the Atmega and the RPi are in communication using UART. The connections between Atmega and RPi can be seen in figure (). The Atmega sends accelerometer data, and keypad inputs to the RPi which is displayed on screen. The RPi sends commands to the Atmega when the door needs to be unlocked which is when the Atmega rotates the motor. 

* Focus on the technical details in this section. Include all schematics, drawings, pictures etc. Each object mentioned in 'Architecture' should be explained here with:
  + The implementation of the object: electrical, programming, mechanical
  + Signals coming in: what type of signal, protocol used (if any), interfaces, electrical parameters etc.
  + How the object is implemented in details:
    - If you have the schematics, provide it here
    - If you have the algorithm that you will use as object implementation – provide it here
* If possible, provide schematics (or other implementation) for both high and low level objects.
* If you have PCB design, include it in this section along with the corresponding schematic.

**Testing**

The CNN(FaceNet) was the only major competent that required testing. With a lack of physical testing, KED was tested using a database of faces. KED was tested with Labeled Faces in the Wild which is a public benchmark for face verification. FaceNet was then feed several of these images, and it outputed the results of the CNN. While it was not tested for all the faces in the database, FaceNet was accurate, and it was able to recognize authorized users. There were serval pictures that fairly close to be being determined as an authorized user.

Provide the description of how to test your device, when it will be finished

* Inputs / actions to the device
* Expected outputs or reaction of the device

If you did any tests (either tests of prototype, or just some component) – include test procedure and results in this section.

**User's manual**

Write the user's manual for the end customer: how to setup, use, how to connect, etc.

**Step 1**

Connect the motor, keypad, RPi, and reed switch to the appropriate ports on the printed circuit board

**Step 2**

Connect the Camera and AC adapter to the RPi. Connect the RPi to a monitor/TV. Power up the RPi

**Step 3**

Either take a frontal picture of your face with included camera or import a picture and save it in the KED folder on the RPi. Make sure the picture of your face is well lit with even lighting.

**Step 4**

Close out of the KED folder and open a terminal on the RPi. Navigate to the KED folder and run the KED main program

**Step 5**

After KED boots up, the main menu is displayed on the monitor. Displayed are different actions that can be performed.

**Step 6**

For set up, navigate to “ARM/DISARM the security.” This toggles the security of KED. If the disarmed, KED will not send an email alert to you in case of a break in. It is strongly advised to keep this option on.

**Step 7**

Navigate to “Generate Key” option. Selecting this option will open the key generator menu. Keys are another option of unlocking KED. Keys have an expiration date. You can select what time and how many days the key will be available. The type of key can be also changed here like if this a key for a friend/family or for type of courier. If the numbers in the key is not to your liking, the key’s numbers can be randomized with the regenerate command. 10 keys are the limit that you can use.

**Step 8**

Exit the Key menu. Navigate to the “Settings” option. The sensitivity of KED can be changed here. Higher Sensitivity makes it easier for KED to recognize your face, but it is more likely to trigger false positives. Accelerometer sensitivity can also be changed here. Higher Sensitivity leads to can detect break ins better at the cost of more false positives.

**Roles & skills in the project**.

|  |  |  |
| --- | --- | --- |
|  | **Objects involved** | **Required skills** |
| **Microcontroller programmer** | * Atmega328p * Stepper Motor * UART | * Knowledge of ATMEGA168 microcontroller * C++ programming * AVR Studio experience |
| **CNN programmer** | * Image Processing | * Knowledge of Neural Networks |
| **PCB design** |  | * Knowledge of Eagle |

Table 3. Roles & skills

|  |  |
| --- | --- |
|  | **Assignment** |
| **Microcontroller programmer** | Bryan Takemoto |
| **CNN programmer** | Adrian Ruiz |
| **PCB designer** | Bryan Takemoto |

Table 4. Roles assignment

**Parts list**

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part type** | **Vendor** | **Model** | **Parameters** | **Picture** | **Att. id.** |
| **Primary Microcontroller** | Raspberry Pi | 4 | Central component to perform facial recognition. |  |  |
| **Secondary Microcontroller** | Microchips | ATmega328/328p | Manages accelerometer and motor. | A circuit board  Description automatically generated |  |
| **Camera** | Raspberry Pi | Camera Module V2-8 Megapixel (1080p) | Captures images for facial recognition. |  |  |
| **Accelerometer** | HiLetgo | MPU6050 | Measures force applied to the door. |  |  |
| **Voltage Regulator** | MCIGICM | L7805 | Provides a constant 5V DC to the circuit. |  |  |
| **Power Adapter** | N/A | 9V 1.5A Power Adapter | Converts the AC to 9V DC which is fed to the voltage regulator. |  |  |
| **Stepper Motor** | STEPPERONLINE | Short Body NEMA 17 Bipolar Stepper Motor | Controls the lock’s position. | A picture containing electronics  Description automatically generated |  |
| **Motor Driver** | PIXNOR | L293D | Provides enough current to drive the motor. | A sign on the side of a building  Description automatically generated |  |
| **Motor Mounting Hub** | CUSCUS | 5mm Universal Mounting Hub | Mounts to the motor’s shaft. | A close up of a device  Description automatically generated |  |
| **Level-Shifter** | HiLetgo | Logic Level Converter Bi-Directional 3.3V-5V | Safely allows serial communication between primary and secondary microcontroller. | A circuit board  Description automatically generated |  |

**Project timeline**

Include general past project timeline – what happened when

* Add any interesting information
* Add the delays if solving the problems causing delays was interesting

**Final remarks**

Include anything related to the project that you want to add/mention.

**References**

[1] B. Marshall, The study of surveillance cameras,

**Marketing flyer**