ECE/COE 1896

Senior Design

Facial Recognition Lock System with Anti-theft- Conceptual Design

Team 2

Prepared By: Dane Krall

Rayan Hassan

Jake Smith

Table of Contents

Table of Figures

Table of Tables

1. Introduction

The market has grown more technologically advanced with cameras starting to be added to doorbells. For example, the Ring doorbell is a product that homeowners are starting to buy for their protection and safety from potential intruders. It uses facial recognition, which has become widely popular in lock devices due to its high accuracy and efficiency.

The proposed device is a Facial Recognition Lock System with Anti-Theft able to verify a user and unlock a solenoid lock. The system will comprise three main units; the hardware consisting of a microprocessor, camera, lock, power devices and sensors, a Computer Vision Unit responsible for verifying the user using machine learning algorithms, and a mobile application that helps the user to log in and receive warnings about any intruder.

The device will take a video as input and will first check whether the user is real or fake. That is, if the input is a real, live person or an image or video placed in front of the camera to pass as the real user. A Liveness Detection Unit (LDU) will be responsible for that. After the user is identified as “real”, the Facial Recognition Unit (FRU) will capture a live image and feed it to the machine learning algorithm that will verify the user. If verified, the solenoid will unlock, otherwise it will stay locked and warnings will be sent to the real user through the mobile application. Additionally, hardware components like LEDs and a vibrator sensor will turn ON signaling a fraudulent attempt. An email will also be sent to the user to notify them of the attack.

The user will first be asked to login to the mobile application and take multiple face images that will be stored in a dataset. The latter will be fed to the Computer Vision Unit to be used as input to the FRU. In case of any intruder attack, warnings as well as the scammer’s picture will be sent and stored in that dataset. The communication between the app and the other units will be based on HTTP requests which offer lower memory usage.

The device will function under power outages to make sure users are always protected. For that, a backup energy source will be used to make sure it is always powered. The backup source will activate once input power is disconnected or lost. The backup power supply will remain active until its energy supply is depleted or input power is restored.

2. Background

Lock systems have become more reliant on facial recognition, as it is safe to use and provides accurate results. Many machine learning algorithms have been designed for that purpose. Most of them are based on Convolutional Neural Networks (CNNs), as they can process large amounts of data and produce highly accurate predicitons [1]. These models can learn complex and abstract features on their own which makes them extremely reliable. Other algorithms that are widely used are Support Vector Machine (SVM) and K-Nearest-Neighbors (KNN) [1]. The first one maps the data into an N-dimensional space after choosing the N most important features, which reduces the number of features considerably and therefore improves the computational time. That makes it easier for the data to be linearly separable. KNN on the other hand finds the distances between the data points to separate them into classes with the nearest ones being associated to the same class. All algorithms have been proven to produce accurate results in facial recognition. There are many pre-trained models out there that use them.

FaceNet is an example that uses CNN. It was developed by Google and is widely used for facial recognition. It is a 22 layers deep neural network that trains its output to be a 128-dimensional embedding [2]. The loss function used is called triplet loss that minimizes the distance between the anchor (the live image taken by the device) and positives (other stored images of the same user) on one hand, and maximizes the distance between the anchor and negatives (other training images of random individuals) on the other. The problem with pre-trained algorithms like this one is that they might not be fully compatible or transferable to any task since they are trained using a fixed dataset that may not resemble the intended one [3]. Therefore, problems like image sizing and number of classes can be encountered. Additionally, they are too complex to understand and debug.

The proposed approach uses a simple machine learning algorithm. The first design concept uses a CNN algorithm that can be trained using any dataset with any dimensions, since it resizes all the input and ensures consistency among all training and testing images. Also, it uses fewer layers and clear, changeable kernels and activation functions. Every element can be adjustable to suit any specific task. The second design concept uses SVM with Principal Component Analysis (PCA). PCA is a dimensionality reduction method that “summarizes” the content in a large dataset into a smaller one that retains only the most important features and handles the data in a lower dimensional space. Either one of these design concepts will ensure adaptability and ease of use, as well as high accuracy and better computational time.

As facial recognition is becoming more popular with the years, another problem arises. Spoofing is when a scammer disguises themselves as a known or trusted source. This can be done through holding a picture or a video of the real user in front of the camera. Many solutions have been implemented to solve that problem like the use of local binary patterns (LBPs), a method that studies the surrounding pixels of a point to analyze the textures of the image in more detail [4-5]. This method is usually combined with Difference of Gaussian (DoG) and SIFT to classify the input as live or spoof [5]. These methods enhance the edges of the noisy images and further detect and analyze features respectively. Although this approach produces a high level of accuracy, it performs poorly with specific, unconstrained scenarios because the features are sensitive to variations in lighting. Other anti-spoofing methods include CNN algorithms that fuse different frames of the video taken to study their respective features and detect any changes in the edges of the picture to figure out if the scammer is holding a picture or not [4]. This method is too complex and might not always work if the picture held in front of the camera is still for example.

To solve that problem, an interactive mechanism will be implemented in the first design concept. The user will have to perform certain movements following two LEDs. The first one will prompt them to tilt their head to the left and the second one to the right. The LEDs will turn ON randomly so that the scammer doesn't know what to expect and therefore can't have a pre-prepared video of the real user performing these actions. Three checks (three movements) will be made to ensure a higher level of accuracy. The second design concept proposes a blink detection mechanism that will make sure the user is a live person by blinking every few seconds and therefore prevent spoofing using images of real users.

On another hand, there needs to be a way to interface with the locking system to make it easier for users to track fraudulent attempts and manually control the lock. Many lock systems have app interfaces such as Lockin, SMONET, Veise, and ULTRALOQ. These take advantage of the ever growing ubiquity of these devices like how stated in Objective, a widely known tech blog. The amount of smartphone users in 2020 was expected to reach 3.8 billion [6]. This simplifies designs and cuts extra costs. The problem with the majority of these is that they provide a simple feature set.

The proposed device will have features found in higher end models like a login system or security alerts. The lock will also use HTTP requests to transfer data which is better since it only allows connection when needed, which decreases cyber attacks [7].This will also allow for a system with less latency by cutting out handshake procedures [7].

Finally, consistent power flow to activate a device to remain operational at all times is an issue that needs to be addressed. Power outages have become an issue due to the rise of electric grids switching over to renewable power that is not as stable compared to using fossil fuels or nuclear fission. As a result, not all utilities are able to consume electricity that is needed to keep items active at all times. For example, California has rolled out solar power extensively and has suffered through blackouts as a result of renewable energy [8]. The electrical grid will be seeing changes in the upcoming years as renewable energy starts to grow more prominent as fossil fuels continue to dwindle later this century. Renewable energy is not completely reliable yet as it is not as stable as using fossil fuels to create electricity.

A backup energy source needs to be accounted for. There are devices around the world that need to be powered up 24 hours a day and 7 days a week. In the event that input power is lost, a backup power supply needs to be incorporated into the system to keep components active [9]. The backup power supply will serve as a counter to the electrical grid as countries across the world begin to grow accustomed to alternative forms of generating electricity.

3. System Requirements

3.1 Functional Requirements

3.1.1 Liveness Detection Unit

- The system shall not start before a face is positioned in front of the camera.
- The system shall take the live video as an input and produce a binary output of whether the user is real or fake.
- The system shall detect the live features of the user’s face such as eye or head movement.
- The system shall alert the user if spoofing is detected using LEDs, a vibration sensor and warnings sent to the mobile app.
- The whole process shouldl not take more than one minute to execute and finish.

3.1.2 Facial Recognition Unit

- The system shall start right after the LDU and only if the user is identified as “real” (or “live”).
- The system shall take 3 inputs: the live image taken of the user, positives (other images of the user stored in the dataset) and negatives (images of random people).
- The output of the system shall be binary; class 1 (verified) and class 2 (unverified).
- The system shall notify the user if verified or not using LEDs and a vibration sensor.
- The system should not take more than one minute to verify the user.

3.1.3 Data Transmission Unit

- The system shall only let authorized users access and modify the data being used in the locking system.
- The system shall transmit pictures taken from the app to the FRU to register valid users.
- The system shall transfer a signal to the solenoid lock system allowing it to be unlocked without use of the FRU in case of manual control mode through the app.
- The system shall notify the user of attempted illegal entry through use of the vibration sensor and three or more invalid attempts of the camera.
- The system shall keep a record of who used the system.
- The system shall have a login system.

3.1.4 Power Backup Unit

- The system should take in 120VAC for input power from a wall outlet.
- The system should revert to backup power in the event that the input power source is disconnected or lost.
- The system should use the rectifier to step 120VAC down to 5VDC to power the module.

3.2 Non-Functional Requirements

3.2.1 Liveness Detection Unit

- The algorithm should use the openCV library in Python.
- A monitor should show the video taken of the user at any time where the programmer can track the eye position or head movements with live coordinates.
- The algorithm shall be robust and produce high accuracies of liveness detection

3.2.2 Facial Recognition Unit

- The algorithm should be written in Python and make use of libraries like scikit-learn and openCV.
- The system shall use a robust machine learning algorithm that produces high accuracies of 90% or more.
- All input images shall be of a consistent size (for example, 128 x 128 pixels).
- The data should be split into 80% for training and 20% for testing.

3.2.3 Data Transmission Unit

- The app should be written in Android Studio using either C++,Java,Javascript, or Kotlin.
- None of the app functions should take over 3 seconds to complete.
- The data should be sent using HTTP requests to a central server.

3.2.4 Power Backup Unit

- The system should be wired and soldered in a way that prevents the user from facing any electrical hazards.
- The system should be at full power to increase longevity if input power is lost.

4. Design Constraints: Standards and Impacts

4.1 Conventional Constraints

4.1.1 Time

Time is one of the biggest limitations for this project since the team only has 12 weeks to come up with a fully functional design. Multiple measures will be taken to overcome that like planning a fixed and detailed schedule, laying out roles for each member and having good communication. This will allow the team to be in synchronization and consistent throughout the semester. Enough time must be allowed to complete the prototype and perform testing.

4.1.2 Manpower

The team will consist of only three members. Therefore dedication and communication are required from each member to ensure the project is successfully finished in time.

4.1.3 Financial

The budget for the project is 200 US dollars. Any component needed should be thoroughly discussed and researched, in order to ensure that the team stays within the budget and still gets all the necessary tools. The team should provide clear reasons and explanations in case the budget is to be surpassed.

4.2 Project Constraints

4.2.1 Computer Vision Related Constraints

These are limitations related to images and videos taken of the users.

- The system will not accommodate extremely dark or extremely light setups. Captured pictures should be clear enough for the algorithm to work accurately.
- The user should be standing still in front of the camera. If the input video/image is flu, the algorithm might produce wrong results.
- The algorithm will not be designed to specifically adapt to facial changes like beards, glasses, masks, etc. The training images will consist of a variety of face pictures but will not guarantee variety in facial features (of the same person).

4.2.2 Data Transmission Constraints

- The system will not entertain the effects that a large number of users could have on transmission speeds
- While the system will be easy to use and made as simple as possible, basic technological literacy will be expected to use our product.
- The data transmission rate is determined by the Wi-Fi standard used by the router.

4.2.3 Power Constraints

- The system will not last more than one hour after being disconnected from input power. The available backup power is limited based on the size of the supply being used in the system.
- The system will be wired in a way that prevents users from getting electrocuted in the event that the PCB is exposed by the user. The AC power flowing into the rectifier will need to be wired and soldered so that no stray wires are exposed.

4.3 Impacts in Non-Technical Contexts

4.3.1 Environmental

The backup power supply may have environmental issues that are not limited to carbon emissions, air pollution, noise pollution, or resource consumption (“Backup Systems”). An uninterruptible power supply (UPS) will run into noise pollution as it starts up when input power is lost. In addition, the batteries inside of a UPS will need to be replaced over time.

4.3.2 Public Health

The device must be wired so that the National Electric Code (NEC) is followed for the respective location. In the United States, each state follows a different edition of the NEC.

4.3.3 Global, Cultural and Societal

Increased ubiquity and accessibility of lock systems can be a deterrent to those who would perform thefts. This could decrease the rate of spontaneous thefts that occur in society making a safer environment for individuals.

4.3.4 Diversity, Equity and Inclusion

The device can be used by anyone, from any different age, race, ethnicity, etc. The application will make it accessible and easy to use for anyone. Steps will be clear for first time users on how to login and manually control the lock. This will ensure that even people with accessibility issues can easily use and understand it.

The computer vision unit and hardware will walk the user through a bunch of interactive steps to ensure authentication (whether the user is real or fake) and verification. These will be mainly based on LEDs and a vibration sensor. This will ensure that even people with hearing problems can use it. Additionally, the system won’t require any physical movement by the user, all they have to do is position their face in front of the camera. This will prevent any issues for people with certain physical disabilities.

4.3.5 Welfare and Safety

The device will prevent spoofing; that is, assuming someone else’s identity by placing a picture or a video in front of the camera. The system will detect it and send warnings to the real user. Additionally, the lock won’t unlock unless the user is verified, which will be based on a robust machine learning algorithm for facial recognition. That way users will be safe and protected against any intruder attack.

4.3.6 Economic

If the anti-theft system was installed to serve as a lock system for a business or corporation, the business would benefit by not having to deal with robberies. The use of the system would allow businesses to save money in the event they have to deal with a robbery in a power outage.

5. Conceptual Design

5.1 Device Description

The device consists of an anti-theft system based on facial recognition. It will comprise three main sections, a mobile application that helps users to log in or manual control the lock, a computer vision unit responsible for verifying the user as well as preventing spoofing and a hardware unit which holds the system together and includes sensors, power devices and a camera.

5.2 Design Concept 1

5.2.1 Software Design

The software design is split into two parts. The Application Unit that will be developed for the user to login, manually lock or unlock the device, and get notified of any failed attempts. It will also hold pictures of the user in a dataset that will be fed to the machine learning algorithm for training. The other part is the Computer Vision Unit composed of a Liveness Detection Unit (LDU) for anti-spoofing purposes and a Facial Recognition Unit (FRU) that verifies the user. The following figure shows the interaction between the software components.

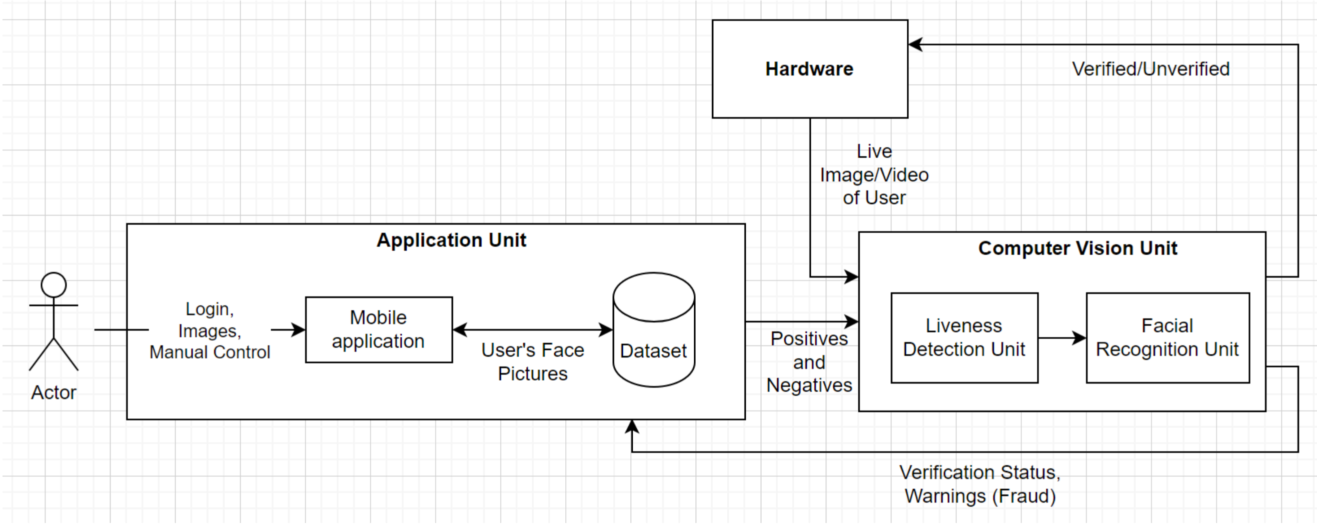

Figure 1: Block diagram for software design

5.2.1.1 Data Transmission Unit

User Interaction

The primary option is using a mobile app to interface with a Raspberry Pi to send signals to the locking system. The app would allow the user to create profiles for the locking system to recognize. The app would also keep a log of people who have opened the lock system with a timestamp of when it was used in our database. The system will also allow the user to control the lock manually through the app in cases where the FRU might malfunction or the face is covered. The app would also send notification of illegal entry either due to tampering with the use of the vibration sensor or after three invalid entries. The app will be constructed using Android studio which allows emulation of the app and will be written in C++.

Log In

When first entering the app, the user will have to create an account with a username, email and password. A code will also be required to connect a specific lock system to your account.

Creating Profiles

When creating profiles the user will be asked for their name and be required to take pictures of themselves to be held in the dataset. The name information will be used in the entrance log of the system to display a record of users and times. The pictures will be used in the computer vision program to know valid users to detect.

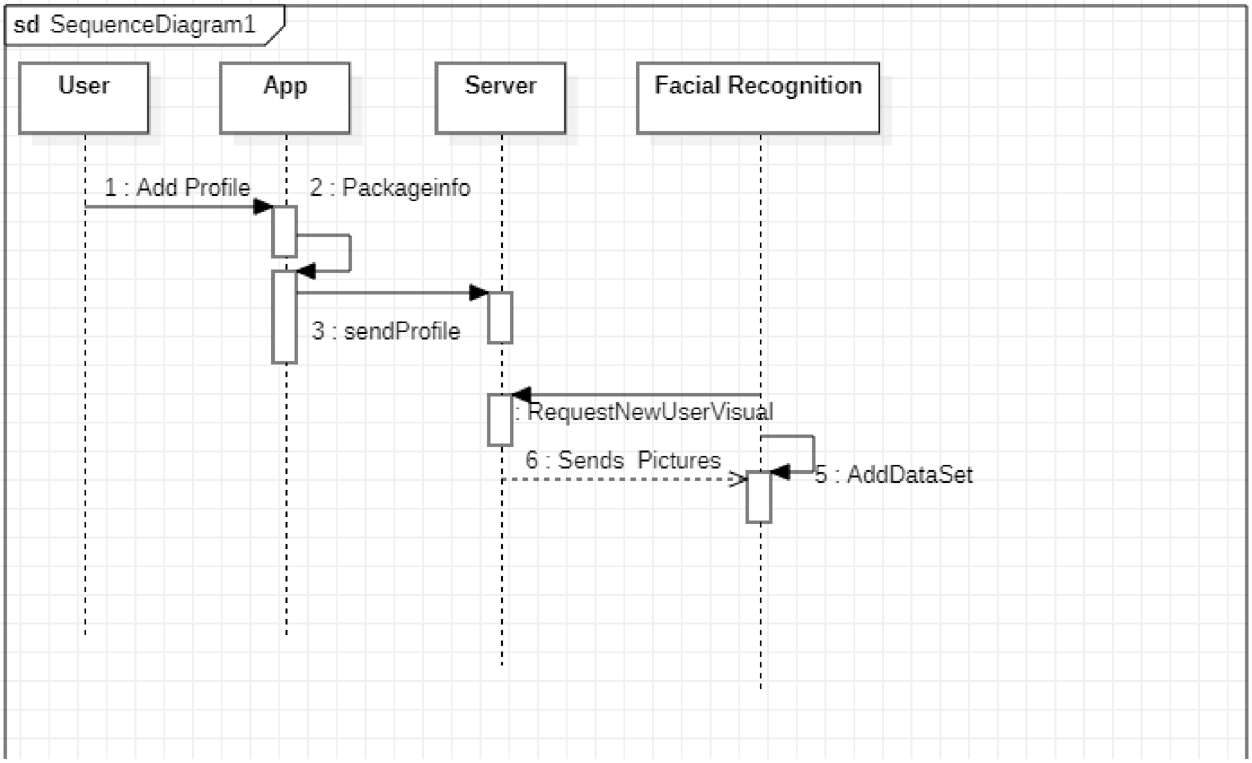

Figure 2: Sequence Diagram for user interaction

Lock Interface

The lock interface of the app will be a simple button that will send a signal to the Raspberry Pi changing necessary GPIO values to unlock or lock the system. The system will have to have a cooldown between presses of the button until the lock is finished changing positions to avoid interfering signals and breaking the lock mechanism.

Security Alerts

When an unauthorized user tries to open the lock three or more times or the vibration sensor is triggered showing attempted tampering of the device a notification will be sent to the phone and by email to the user. The camera will also take a picture of the invalid enterer and take video while in the range of the camera's detection.

Entrance Log

Whenever the user successfully opens the lock using the system, the app will save their information. That is, the name of the given profile that is stored along with the timestamp of when the entry happened.

Time Frame

Actions completed by the app should happen in around three seconds since anything above that has been found to be unacceptable by users of phone apps [10].

Data Set

The Data Set will store all the information needed for the app and computer vision modules. It will store usernames, passwords, timestamps, pictures, videos, and lockstate. Names and passwords will be encrypted for security and privacy purposes against threats and attacks to the system that would try to steal information.

5.2.1.2 Computer Vision Unit

Input Handling

The camera connected to the Raspberry Pi will take a video of the user. Controlling the video will be made possible using the Picamera library in Python. Functions that help with that are:

- camera.start_preview()
- camera.start_recording()
- camera.capture()
- camera.stop_recording()
- camera.stop_preview()

The captured video will be continuously used for liveness detection until the user is marked as "real". After that, an image will be captured, resized and fed to the machine learning algorithm for facial recognition.

Liveness Detection Unit

The LDU will use an algorithm to detect head movement. LEDs will light up one at a time to prompt the user to tilt their head to the left or to the right. After these movements are executed, the user will be labeled as "real", otherwise it will be a "fake" person. First, the algorithm will detect the user's face (head). Two points will be taken from across the face so that it forms a line. These two points can be the eyes' centers. They will have coordinates (x1,y1) and (x2,y2) respectively, where the x-axis is the horizontal axis and the y-axis is the vertical one. The angle between the line formed by the two points and the x-axis will be calculated using the formula [11]:

$$\theta = arctan\left(\frac{x2-x1}{y2-y1}\right)$$

The sign of θ will determine whether the head is tilted left or right. The movement will be classified as left or right provided a margin of error of 15 degrees. So if the face tilts more than 15 degrees on either side it will classify as right or left.

The LEDs will be connected to the GPIO pins of the raspberry pi. Small resistors are also needed accordingly.

Facial Recognition Unit

This implementation will use a Convolutional Neural Network (CNN). It can be divided into three main parts: Data Handling, feature extraction and classification.

Part 1: Data Handling (input)

The image captured by the camera will be reduced to a face image (without the background). For that, Haarcascade implementation will be used to detect edges, lines and directions of the picture which will help to detect the face. For instance, it can easily detect sudden changes in lighting, which helps in knowing where the boundaries are between the background and the face. The haarcascade_frontalface_default.xml file that will be taken from an open source library (opencV) [12]. After that, the face image will be converted to a matrix for input.

Part 2: Feature Extraction

This part will be done following a series of convolutional and pooling layers. Specifically, the algorithm will have 3 convolutional layers using 32, 64 and 128 filters/kernels respectively of size 3x3. With each convolution layer there will be a pooling layer to reduce the spatial dimensions of the matrices. MaxPooling will be used with dimensions 2x2 or 3x3. After those two layers, a flattening layer will convert the feature maps (matrix) into a 1-D vector, which will be the input to the classification layers.

Part 3: Classification

This stage will use the extracted features to assign a probability distribution over a set of predefined classes (different individuals' identities). The flattened vector passes through fully connected layers to help it learn the complex relationships between extracted features and the facial identities. The activation function will be ReLu: f(x) = max(0,x). This function will allow for a more linear behavior of the neural network and it is computationally simple (only makes use of the max() function).

The loss function used will be a Categorical Cross-Entropy Loss (commonly used for facial recognition purposes):

Loss = $-\sum_{i} y_{i} log(p_{i})$, where y is the output vector and p is the probability of each class i.

Training and Testing

The dataset of images will be split into 80% training and 20% testing. The algorithm will be trained with a dataset taken from an open source of images of random individuals (negatives) as well as images taken from the mobile application of the user (positives).

Class diagrams and function calls

The class diagram on the next page shows the dependencies between each class as well as the attributes and main functions calls in each. These might change throughout the semester but the main purpose of each class is fixed and can be understood by the naming of variables and functions.

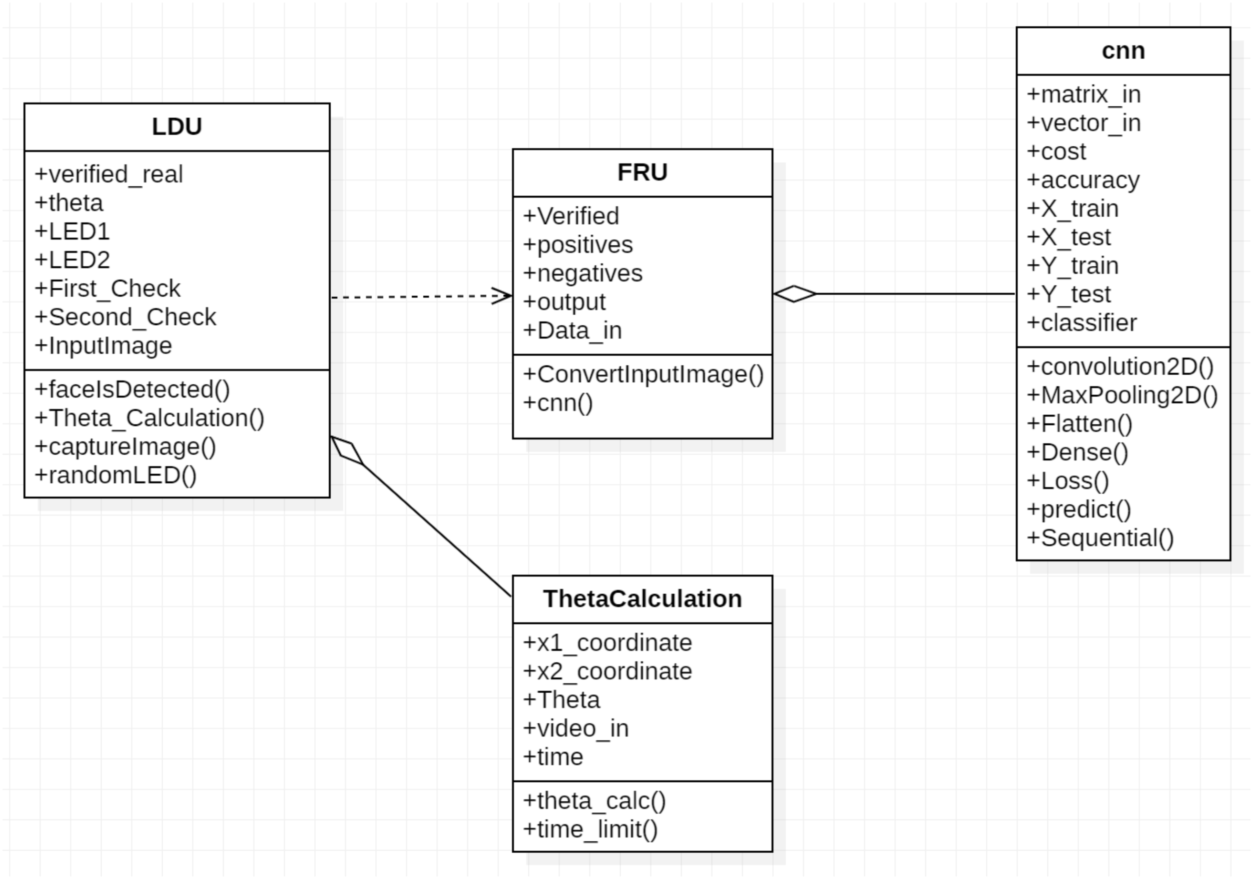

Figure 3: First Class diagram

The pseudo-code of the overall structure of the algorithm is shown below. This is only the “skeleton” of the code; the main functionalities of LDU and FRU (like the machine learning algorithm) will be implemented in the functions mentioned in the pseudo-code.

```
class LDU
while faceIsDetected() and verified_real = None (user not yet identified as real or fake):
    -   Theta = Theta_Calculation()  (from thetaCalculation class)
    -   randomLED()  (this will randomly turn ON LED1 or LED2)
    -   while LED1 is ON
            -   If Theta < -15
                    -   First_Check = True
            -   Else
                    -   First_Check = False
    -   While LED2 is ON
            -   If Theta > 15
                    -   Second_Check = True
            -   Else
                    -   Second_Check = False
    -   If First_Check = True and Second_Check = True
```

```
        - Verified_real = True (user is “real”)
        - Input_Image = captureImage()   (capture image for the FRU)
    - Else
        - Verified_real = False (user is ”fake”)
        - Raise error
```

class FRU

```
If verified_real = True (user is “real”):
    convert the image into a matrix
    Data_in = ConvertInputImage()

    use the cnn() function in the cnn class feeding it the input matrix, and the positives (other
    face pictures of the user), and negatives (face pictures of random people)
    Output = cnn(Data_in, positives, negatives)

    If output of cnn algorithm is True classify as verified, else unverified the user
    If output = True
        - Verified = True
    Else
        - Verified = False
        - Raise an error
```

Class ThetaCalculation

- Calculate the coordinates of the two points x1 and x2
- Calculate theta using the formula
- Keep track of the time using time_limit() (so that the movements are executed within the specified time limit – 5 to 10 seconds each –)

Note that this function will always get called as long as the face is detected in the video and the user is still not identified as “real” or “fake”

Class cnn

The library used will be Keras in Python

- Start by creating the convolutional neural network using sequential()
- Add the convolution layers specifying the kernel and the activation function, and each followed by a maxPooling layer specifying the pool size (use functions convolution2D() and maxPooling2D())
- Flatten the output using flatten()
- Create a fully connected layer using the function dense()
- Calculate the loss function using Loss()
- Train the model specifying the training and testing data set (as well as validation data set if needed)
- Test the model using unseen images and calculate the accuracy

5.2.2 Hardware Design

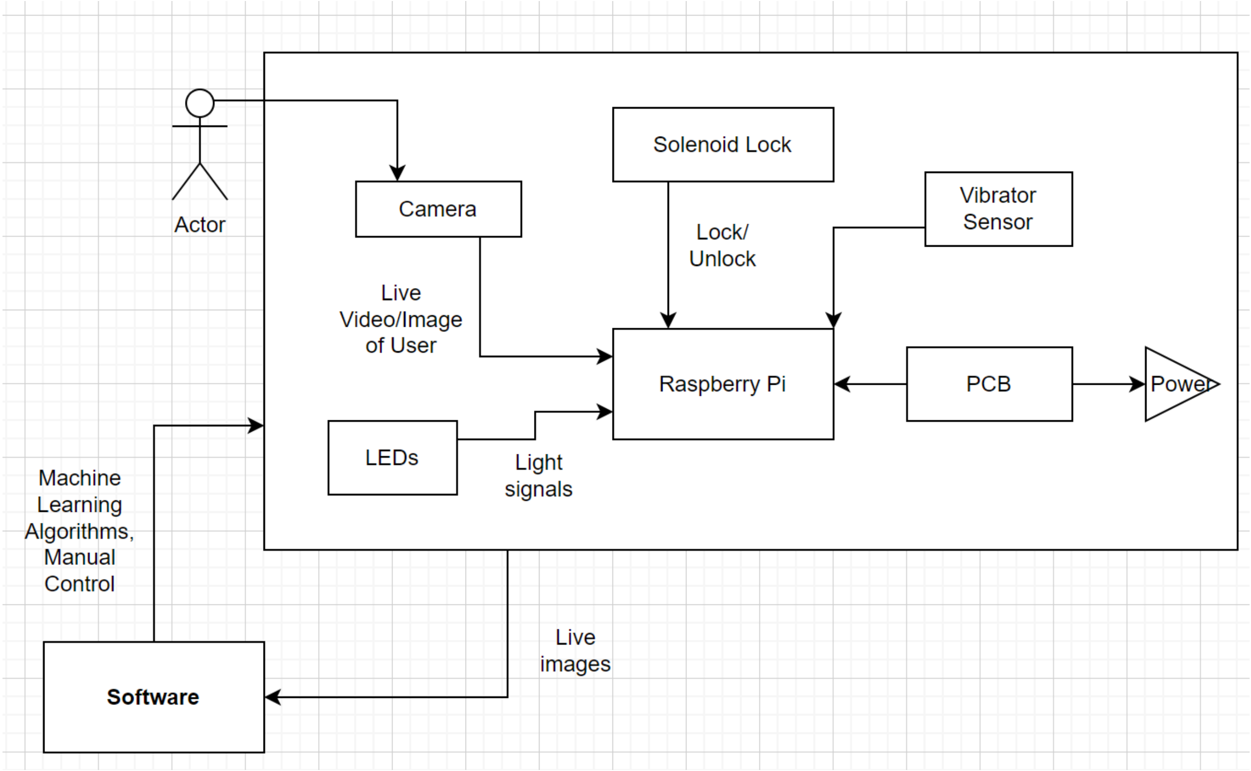

Figure 4: Block diagram for hardware design

The image above is the hardware design that will be implemented for the design of the Raspberry Pi design concept. The image shows how the separate hardware modules will integrate with one another once all of the modules are put together in addition to how the hardware interacts with the software.

Power

The power for the circuit will be 120VAC from a wall power outlet. To allow the circuit to be powered in the event of a power outage, the input power will feed into an Uninterruptible Power Supply (UPS). When the UPS detects an outage from the input power, the UPS batteries kick on and will power the circuit until the power finally runs out. The UPS will only be able to run for a limited time so that input power can feed back into the UPS to power those batteries back up. The system will last no longer than one hour and will serve as a design constraint with the limited power that the UPS is able to supply.

PCB

A PCB will need to be designed in order to integrate all the customized components together for the project. One of the main parts of the design will be to hold the rectifier to power the Raspberry Pi module. The rectifier will take the 120VAC and step the circuit down to 5VDC.

Raspberry Pi

One advantage of the Raspberry Pi is the amount of GPIO ports compared to the Arduino [13]. “The Raspberry Pi has access to 40 GPIO pins and 8 analog pins while the Arduino has 14 GPIO pins and 6-8 analog pins” [14]. This allows for flexibility for the design without having to worry about the available pins.

Vibrator Sensor

The vibrator sensor will be used to detect movement when a person approaches the module. When the sensor detects movement, the camera will then power on. Energy conservation needs to be a factor in the event of a power outage to increase the longevity of the UPS. The sensor will be connected to the GPIO pins of the Raspberry Pi module.

Solenoid Lock

The solenoid lock will lock or unlock based on if a recognized face is detected by the camera module. The GPIO pins on the Raspberry Pi will control whether the lock is in the locked or unlocked position based on the camera detection that communicates with the Raspberry Pi.

Camera

The camera will detect a person’s face to determine if they are a recognized person or not through image processing. The camera will connect directly to the Raspberry Pi in the appropriate input slot. The camera will activate when the vibration sensor detects movement in an effort to conserve power.

LEDs

The LEDs will be used to show the steps of the face recognition process. An LED will show a step awaiting the person to approach the module. Once the vibration sensor detects movement, the camera powers up and a different LED powers up to scan the face. If a recognized face is detected, a green LED will light up. If not, a third LED will power on for an unrecognized face. Other LEDs will be used for the LDU.

5.3 Design Concept 2

5.3.1 Software Design

The overall software design is the same as described in Figure 1, however each sub-unit will be implemented differently.

5.3.1.1 Data Transmission Unit

Web App

A web app will be made to perform the needs of a user in order to interact with the lock system. This functionality would be the same as stated in the phone app being able to login, create profiles, unlock the solenoid and create a record of the device's history. Instead of emails, notifications will be sent as text messages to the user. The web app will be written in Python instead of C++ and be run through an .exe file. The UX of the app would be implemented through PyQT.

5.3.1.2 Computer Vision Unit

Liveness Detection Unit

This LDU implementation is an eye blink detection algorithm. Each eye is represented by 6 points (p1,...,p6) with (x,y) coordinates starting from the left-corner of the eye and working its way clockwise around the rest of the region. The following figure depicts that:

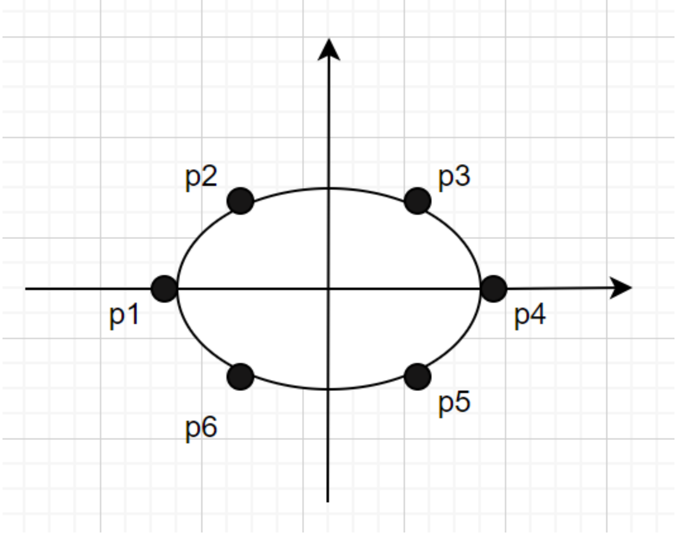

Figure 5: Detection of points around the eye

The circle/oval in that figure represents the eye. After that, an Eye Aspect Ratio (EAR) is calculated [15]:

$$EAR = \frac{||p2-p6||+||p3-p5||}{2||p1-p4||}$$

The EAR is constant when the eye is opened but will decrease close to zero when it is closed. Another way to view it is when the eye is closed, the 6 points will all get closer to the horizontal x-axis. This ratio will be constantly updating until a blink is detected. However, if none is detected after 20 seconds, a warning will be sent and the user will be identified as “fake”. The algorithm will also use openCV and the live frames captured by the camera.

Facial Recognition Unit

This implementation will use Principal Component Analysis (PCA) and Support Vector Machine (SVM) for facial recognition.

Part 1: PCA

The input images are first flattened into a 1-D vector. Each pixel value is normalized to have a mean of 0 and a standard deviation of 1. This helps to scale the features for PCA, which is later used to reduce the dimensionality of the original data while maintaining the most important features. The algorithm goes as follows:

- Calculate the covariance of the image dataset
- Compute eigenfaces and eigenvalues of the covariance matrix
- Sort the eigenvectors in decreasing order of their corresponding eigenvalues
- Select the top K eigenvectors (those retain the highest variance, which means they are the most important features) to retain K principal components (PC).
- Project the preprocessed images onto the K PCs to obtain a lower-dimensional representation of the data

Part 2: SVM

This classifier will find a decision boundary to best separate the data points belonging to different classes in a higher-dimensional feature space. The hyperplane should maximize the distance between the nearest data points of different classes. Packages for SVM will be used from libraries like sckit-learn in Python.

Training and Testing

This implementation will also split the dataset into 80% training and 20% testing.

Class diagrams and function calls

The class diagram on the next page shows the dependencies between each class as well as the attributes and main functions calls in each if this implementation is to be used.

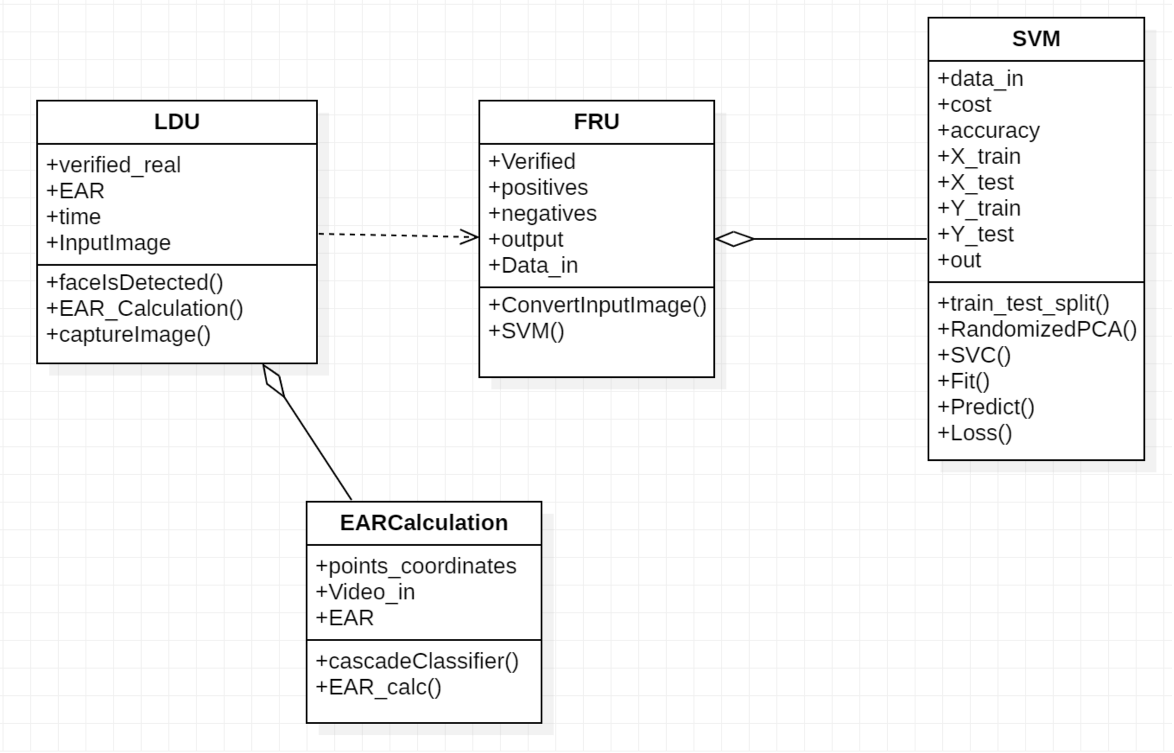

Figure 6: Second Class diagram

The pseudo-code of the overall structure of the algorithm is shown below. As mentioned before, this is only the “skeleton” of the code.

class LDU

(EAR here will be a list storing multiple values of EAR within the time limit given to the user to blink, which will be 10 seconds)

Start counting seconds using time()

```
while faceIsDetected() AND verified_real = None (user not yet identified as real or fake) AND time<10s:
    -   EAR.append(Theta_Calculation())  (from EARCalculation class)
        -   If the length EAR is higher than 2 (so when it had time to store multiple eye movements)
            -   if min(EAR)<<max(EAR) (to ensure that eyes were opened and closed):
                -   Verified_real = True
                -   InputImage = captureImage()  (capture image for the FRU)
                -   break
            -   Else:
                -   verified_real= None  (this will stay none to give more time to the user to blink, other it will directly break the loop if it is False)
```

```
                    -
After 10 seconds, if verified_real is still None:
    -   Verified_real = False
    -   Raise error
Else:
        (nothing happens, verifier_real stays True and algorithm proceeds to FRU)
```

class FRU

```
If verified_real = True (user is “real”):
        convert the image into a matrix
        Data_in = ConvertInputImage()

        use the svm() function in the cnn class feeding it the input matrix, and the positives
        (other face pictures of the user), and negatives (face pictures of random people)
        Output = svm(Data_in, positives, negatives)

        If output of SVM algorithm is True classify as verified, else unverified the user
        If output = True
            -   Verified = True
        Else
            -   Verified = False
            -   Raise an error
```

Class EARCalculation

- Locate the eyes of the user (use cv2 library and CascadeClassifier() function)
- Calculate the coordination of point p1,...,p6
- Calculate EAR using the formula

Note that this function will always get called as long as the face is detected in the video, the user is still not identified as “real” or “fake”, and the time is less than 10 seconds (which is the max amount of time given to the user to blink)

Class svm

The library used will be sklearn in Python

- Arrange and resize data (matrices)
- Split data into training and testing using train_test_split()
- Use function RandomizedPCA() for dimensionality reduction of data
- Use SVC() function for SVM and test for different kernels
- Train the model using training dataset
- Test the model using testing dataset (predict() function)
- Calculate cost function and accuracy

5.3.2 Hardware Design

Power

The power for the circuit will be 120VAC from a wall power outlet. The power will flow into batteries in an effort to keep the circuit powered if input power is lost. In the event of an outage, the batteries should remain powered through the use of solar panels. The batteries can either take in input power from a power outlet or from solar power to power the circuit.

PCB

A PCB will need to be designed in order to integrate all the customized components together for the project. One of the main parts of the design will be to hold the rectifier to power the Arduino Uno module. The rectifier will take the 120VAC and step the circuit down to 5VDC. Additional PCBs may be considered if needed to hold LEDs or needed components.

Arduino Uno

For the GPIO layout, the Arduino Uno has a total of 14 digital pins in addition to 6 analog pins. In addition, the microcontroller has a few slots to supply 3.3VDC and 5VDC power with ground pins. A USB connection is available to program the microcontroller.

Button

The button will be used to detect movement when a person approaches the module. When the sensor is pressed, the camera will then power on. Energy conservation needs to be a factor in the event of a power outage to increase the longevity of the power remaining in the batteries. The sensor will be connected to the GPIO pins of the Arduino Uno module.

Solenoid Lock

The solenoid lock will lock or unlock based on if a recognized face is detected by the camera module. The digital pins on the Arduino Uno will control whether the lock is in the locked or unlocked position based on the camera detection that communicates with the Arduino Uno.

Camera

The OV7670 camera will detect a person’s face to determine if they are a recognized person or not through image processing. The camera will connect directly to the Arduino Uno in the appropriate input slot. The camera will activate when the vibration sensor detects movement in an effort to conserve power. The camera will need to be connected to a serial port reader to analyze the output image.

LCD Screen

An LCD screen will show the commands that must be followed instead of using LEDs to make the project more user-friendly. Code will be needed to verify that the LCD screen is working properly.

5.4 Selected Design Concept

5.4.1 Software Design

5.4.1.1 Data transmission and Application Unit

The design chosen for the Application is the one presented in the one presented in design concept 1. This is a phone app that will interface with both the Computer Vision Unit and the Raspberry Pi. One reason for picking a phone app over a web app is the ease of taking pictures with a phone. A web app might not have a webcam on the computer while most phones have cameras. Another disadvantage of using a web app is if there was a problem with the facial recognition software, a computer would not be as easily accessible compared to a phone to unlock the system [6].

5.4.1.2 Computer Vision Unit

LDU

The design chosen for the LDU is the one presented in design concept 1. It will be an interactive system that prompts the user to perform certain tasks. The reason behind this is that an interactive system ensures a high level of accuracy in liveness detection since it relies on the user’s performance of random head movements in reaction to the LEDs. Therefore, they won’t be able to prepare for it in advance and cheat the system. That way, not only can it prevent spoofing using an image in front of the camera, but also a video of the real user. The random aspect of it is what makes it much more efficient compared to the second design concept which only relies on eye blink detection and fails to detect fake videos.

FRU

The chosen implementation is the one that uses CNN. The use of a neural network is more efficient as it learns complex features on its own in order to make better predictions. Additionally, its built-in convolutional layers reduce the high dimensionality of images without losing its information. Finally, CNN is more suited for large datasets, which is what facial recognition requires, as opposed to SVM.

5.4.2 Hardware Design

Raspberry Pi

The microcontroller that we decided to use for the project is a Raspberry Pi. Programming will be able to be performed directly onto the module. In addition, there is an input available for the camera to be plugged in which the Arduino Uno does not have. The wiring for the Arduino Uno camera requires more precision and time to obtain a stable image. The Raspberry Pi allows for more processing power compared to the arduino [13]. This is advantageous for the facial recognition software that needs to be run on it and the communication between it the Raspberry Pi and the app. This also allows storage for the information that will be sent between the modules.

In addition, the camera that will be used for the Raspberry Pi provides a more stable video feed compared to the Arduino Uno camera. The Raspberry Pi is easier to work with in regard to machine learning and computer imaging capabilities.

LEDs

LEDs will be used for the scanning process over the use of an LCD screen. One concern that came up with using an LCD screen was power consumption in relation to using the LEDs. In the event of an outage that required backup power to be used, the circuit needs to be powered for as long as possible.

Solar Powered Battery Supply

In the event of an outage of input power, a battery supply will be available for use. It will have the ability to take in solar power to last longer than just off of rechargeable batteries. In the event that the design fails or if it may lead to damaged equipment, an Uninterruptible Power Supply (UPS) will be used as the power backup if input power is lost.

6. System Test and Verification

6.1 Software Systems

6.1.1 LDU Testing

6.1.1.1 LDU Function Testing

A code will be run to test multiple functions in the LDU. Some of these tests will be done automatically throughout the semester.

Face Detection

The faceIsDetected() function will be tested using random faces under different setups (lighting, distance from camera, etc.). The expected output is a rectangle appearing around the face of the user (or eyes if following Design Concept 2) signaling that the detection was successful.

LED testing

The control of LEDs used will be tested using GPIO functions in a python script to test communication as well as the LEDs themselves.
To force the LEDs to turn ON or OFF, the function GPIO.output() will be used. The expected output will be ON when the argument to the function is GPIO.HIGH and OFF when it is GPIO.LOW.

6.1.1.2 LDU Overall System Testing

Two tests will be done following the first design concept. First, an image of the user will be placed in front of the camera (common to both design concepts). The LDU should classify it as “fake” and the red LED as well as the vibration sensor should turn ON. In addition, a warning should be sent to the Mobile Application. The second test will be a video placed in front of the camera showing the user’s face. The interactive LDU system in Design Concept 1 should detect an absence of reaction from the user to the head movement requests, which will prompt the system to raise an alert in a similar way.

6.1.2 FRU Testing

6.1.2.1 Time Testing

The time taken by the machine learning algorithm to execute should be tested using the time() function in the time library. The expected time taken should be only a few seconds. If it takes a lot of time, test the algorithms with different kernels, input sizes or activation functions.

6.1.2.2 FRU Function Testing

Start

Test that the FRU won't execute as long as the user is identified as "fake", in other words as long as the output of the LDU is False.

Algorithm testing

The algorithm should be tested using unseen images of the user by brute force (without reading input from camera). The cost is expected to be low (close to 0) and accuracy is expected to be 90%.

Signals testing

Test the communication between the FRU and the mobile app as well as the hardware. The dataset stored in the mobile app containing the user's pictures should be sent to the FRU. Additionally, warnings raised by the FRU in case the user is unverified should be sent to the mobile app. Similarly, these warnings should prompt the red LED to turn ON as well as the vibration sensor.

6.1.2.3 FRU Overall System Testing

The FRU should be tested with multiple users. The input coming from the LDU (capturedImage() function) should be fed to the algorithm and accurate verification results are expected.

6.1.3 Overall Computer Vision Unit Testing

The LDU and FRU should be tested together.

Test cases	Expected outcome
Real input, real user	User classified as "Real" and "Verified". Green LED ON
Real input, fake user	User classified as "Real" but "Unverified". Red LED and vibration sensor ON. Warnings sent to the mobile app
Fake input, real user	User classified as "Fake". Red LED and vibration sensor ON. Warnings sent to the mobile app
Fake input, fake user	User classified as "Fake". Red LED and vibration sensor ON. Warnings sent to the mobile app

Table 1: Test cases for the overall Computer Vision Unit

Note that here “input” refers to what is shown to the camera (i.e whether it is an image or video of the user, or the actual/live user himself).

6.1.4 Data Transmission Testing

The second stage of testing is checking that the data is sent between modules properly and is accurate. This will be done using Post() and Get() requests. The test will also check if the data is correctly stored. Finally, the rate (in MB/s) at which data is being transferred will be tested to determine if data changes at an acceptable rate.

6.1.5 Application Unit Testing

The functions that will be tested are as follow:

- Creating a profile using username, password, email
- Taking pictures of the user to send to the Computer Vision Unit
- Manually control the lock
- View images of scammer attempting to lock the solenoid
- Monitor specific entries (like time of login, time of use, user’s profile, etc.)

6.2 Hardware Systems

Power Testing

Testing will be done to verify that the backup power supply is receiving power from the input power source at 120VAC. Validation is needed for the rectifier to see if the appropriate DC voltage of 5VDC is being generated to power the module. Afterwards, the input power will be disconnected to verify that the backup power system will activate to continue powering the circuit. After verifying, input power will be reconnected to continue the next set of tests.

Component Testing

All hardware components will be checked to verify proper operation. The Raspberry Pi must start after completing the power testing section and boot to the appropriate screen. The vibration sensor functionality will be tested to detect motion and an individual test must be completed for it. A correct response will be generated by an LED. Camera operation will be checked on the module to ensure proper communication with proper video output. The solenoid operation will need to be tested to verify the locking and unlocking motions.

7. Team

The following section describes the background, the skills learned in ECE coursework, and the skills learned outside of ECE coursework for all members of the team.

7.1 Jake Smith

Jake will serve as the electrical engineer of the group for the project. He has design engineering experience with Mitsubishi Electric. He will be responsible for the hardware design. The hardware design includes component selection from third-party vendors (i.e. Digikey) for the microcontroller or microprocessor, the solenoid lock, the vibration sensor, the power system, and the camera configuration. For ECE design, the plan is to design the rectifier to power the Raspberry Pi module, design the vibration sensor to integrate with the Raspberry Pi module, configure the camera to work with the microcontroller, and working with the solenoid to integrate with the team to lock/unlock. Jake will also focus on how to provide power to the system in the event of a power outage at any time. As needed, Jake will design the PCB(s) to integrate shelf and external components together.

7.1.1 Skills Learned in ECE coursework

Jake has taken classes in ECE 0101 - Linear Circuits, ECE 0102 - Microelectronic Circuits, and ECE 1212 - Electronic Circuit Design Lab and will use knowledge from these classes for designing the power system for the circuit. Jake has an understanding of C++ from ECE 0301 - ECE Problem Solving with C++ and ECE 0302 - Data Structures & Algorithms if C++ is used on the Raspberry Pi module. Jake learned PCB and through-hole soldering skills in ECE 1895 - Junior Design. Jake got an understanding of uninterruptible power supplies in ECE 1775 - Power Quality to power circuits in the event of power outages.

7.1.2 Skills Learned Outside ECE coursework

Jake has gotten more proficient with surface-mount soldering through his time at Mitsubishi Electric. With PCB design and soldering, he will be able to reduce the size of circuit boards in order to save costs from the overall budget. Jake may consult with either Dr. Robert Kerestes, Dr. Brandon Grainger, or another professor in the power or electronics department to validate the design of the 120VAC to 5VDC rectifier.

Jake has never used the Raspberry Pi module, so he will need to consult with a faculty member or online tutorials to learn how to use the device to integrate components to. In addition, the Raspberry Pi module comes preloaded with Python and Jake has not learned that programming language. If necessary, he would need to consult with ECE faculty in order to learn how to use the programming language. Jake observed Dr. Robert Kerestes using Python before and may be a good resource in the department.

7.2 Rayan Hassan

Rayan is a Computer Engineering major and will work on the computer vision unit of the project.

7.2.1 Skills Learned in ECE Coursework

Rayan took classes like Junior Design (ECE 1895) and Project and Systems Engineering (ECE 1140), where he worked in teams on a “Bop-It” inspired game and a Train Control System respectively. In the first one, he was in charge of the hardware part where he learned skills like PCB design and prototype testing. In the other one, he worked on the Wayside/Track Controller (software). A great deal of knowledge was gained in both classes, like teamwork and project organization, but also technical skills like Git, PyQt, Arduino Uno, etc. Another important class taken was Introduction to Machine Learning (ECE 1395), where he learned different algorithms and concepts for supervised and unsupervised learning (Regression, Classification, KNN, SVM, CNN, Binary Tree Decision, etc.). Throughout his academic journey, Rayan has learned many languages like Python, C, C++, Java, SQL, VHDL and R.

7.2.2 Skills Learned Outside ECE Coursework

Rayan has had some experience outside of the ECE curriculum. He had an online internship at the Chinese University of Hong Kong as part of the Summer Undergraduate Research Program 2022. He worked on Wearable Robots for Ultrasound Scanners, specifically on localization of targeted areas in US scanning using AI (CNN algorithms). Although his research was conceptual (review paper), he plans to use his knowledge about neural networks for this project.

7.3 Dane Krall

Dane will take on the role of data communication between the phone app, the Raspberry Pi and the locking system. Dane is a computer engineering major at Pitt and also majors in Physics from Slippery Rock University. Dane will be responsible for sending the viable photo data from the app to the Raspberry Pi to be compared against with the facial recognition software. This also includes interactions with a server the app and decoding the signals between all three to perform the actions required of these systems.

7.3.1 Skills Learned in ECE Coursework

Dane has taken ECE 1140 (Systems and Project Engineering) which leads to experience in software development and integration between different modules sending data between each other. This also gave experience in using Github and the importance of communication with a team. Dane has also taken ECE 1895 (Junior Design) and is currently taking ECE 1175 (Embedded Systems), giving him microcontroller experience using Arduino and Raspberry Pi respectively.

7.3.2 Skills Learned Outside ECE Coursework

During module implementation, Dane expects to do research about data communication between a Raspberry Pi, a server running on it, and a mobile app. While he has no prior experience in creating a mobile app, there are similarities between the work to be done as mentioned in ECE 1140. The only server experience Dane has is running a Minecraft server.

8. Schedule and Budget Plan

8.1 Project Schedule

Week 1 (Monday 10/2)

Jake: Design and validate 120VAC to 5VDC rectifier to power circuit.
Rayan: Prepare the dataset for the FRU and convert it to matrices of the same size.
Dane: Design simple app UI and start function implementation.

Week 2 (Monday 10/9) CHECKOFF #1

Jake: Implement rectifier on PCB using Altium Designer.
Rayan: Implement the CNN algorithm and test it using random data (brute force if Raspberry Pi is not received by then).
Dane: Finish function implementations and start testing sending data from app to server.

Week 3 (Monday 10/16)

Jake: Prepare for midterm presentation; solder PCB components together.
Rayan: Work on data handling from the raspberry pi. Capture images coming from the camera and reduce them to a face image.
Dane: Complete being able to send data to server and work on receiving data.

Week 4 (Monday 10/23) MIDTERM PRESENTATION WEEK

Jake: Alternative power supply research and construction.
Rayan: Finalize the FRU. System should be able to capture images coming from the camera and verify the user.
Dane: Finish testing receiving data from the server.

Week 5 (Monday 10/30)

Jake: Configure vibration sensor and solenoid with GPIO pins.
Rayan: Start working on the LDU. Work on data handling: video coming from the camera. Detect faces by drawing a rectangle around them.
Dane: Start integration with the Computer Vision module.

Week 6 (Monday 11/6)

Jake: Implement designed power supply and test with current circuit.
Rayan: Implement the LRU. The algorithm that detects the head movements (left/right) as well as receiving data from the LEDs.
Dane: Finish Integration with the Computer Vision module and begin integration with the Raspberry Pi.

Week 7 (Monday 11/13) CHECKOFF #2

Jake: Finish implementing hardware to finalize prototype construction.
Rayan: Work on the communication between the LRU and the FRU. The whole computer vision unit should be entirely functional.
Dane: Complete integration of all systems.

Week 8 (Monday 11/20)

THANKSGIVING BREAK

Week 9 (Monday 11/27)

Jake: Testing and troubleshooting; prepare for final presentation.
Rayan: Testing and troubleshooting. The Computer Vision Unit should be able to communicate with the hardware and the mobile application.
Dane: Testing and troubleshooting.

Week 10 (Monday 12/4) FINAL PRESENTATION WEEK & EXPO

All: Prepare to present at expo; begin final report.

Week 11 (Monday 12/11) FINALS WEEK (FINAL REPORT)

All: Finish final report and submit.

8.2 Project Budget

Below is the initial project budget consisting of parts that were available at Pitt in addition to parts that needed to be ordered initially.

Part	**Quantity**	**Price (USD)**
Vibration Sensor	1	$2.80
Wall Adapter	1	$8.00
Solenoid	1	$5.50
Raspberry Pi 4B	1	$55.00
Camera Module 3	1	$25.00
PCB	TBD	TBD
	Total	**$96.30**

Table 2: Project Bill of Materials

The Raspberry Pi will be the microprocessor that is used for the project. The camera module will connect directly to the Raspberry Pi to provide a video input for image processing. The wall adapter will temporarily be used to power the Raspberry Pi until the rectifier is designed and implemented. The vibration sensor will be used with the GPIO pins to detect motion. The solenoid will be used with the prototype to simulate the locking and unlocking motions being performed by the module. The PCB(s) needed for the project are currently unknown with the prices, but those will be saved for the final report. PCB(s) will be used for the rectifier design and to hold additional components as needed.

In addition, an Uninterruptible Power Supply is being used for the project. One was already available without the need to take away from the overall budget (valued at about $150). It will be used as a power backup until an alternative power supply can be designed.

8.3 Minimum Standard for Project Completion

The minimum standard for this project is a working computer vision system that can detect between real and artificial faces and verify the user at an accuracy of 90% or more. When detected, the solenoid will open. The device should also work for more than 30 minutes when input power is lost. The lock device will also connect to an app through HTTP requests and a Raspberry Pi. The app will also allow for manual control of the lock as well as adding profiles of users of the system.

8.4 Final Demonstration

The final demonstration of the working modules will be as follow:

1. The user logs in to the mobile application using a username, email and a password. The user will be asked to take face images.
2. A user will stand in front of the camera and follow the LDU check. Different scenarios can be shown like using the wrong user or using a picture/video instead of a live person.
3. The LDU will identify the user as real or fake and sensors (LEDs and vibrator) will turn ON accordingly.
4. If the user is identified as real the FRU will verify them. Sensors will similarly turn ON depending on whether the user is verified or not.
5. If not, the warnings and the intruder’s picture will be sent to the mobile application.

Multiple scenarios will be shown and tested in the final demonstration, but the main functionalities of the system should be primarily demonstrated like the use of the mobile application, the facial recognition and verification of the user as well as the correct output accordingly (sensors). Data transmission will also automatically be shown as the whole modules will function together and communicate.

In addition, the final demonstration will be repeated by running off of the backup power supply to ensure that all systems are working and functioning properly similar to the first test when they were completed off of the input power.

References

[1] L. Li, X. Mu, S. Li and H. Peng, "A Review of Face Recognition Technology," in IEEE Access, vol. 8, pp. 139110-139120, 2020, doi: 10.1109/ACCESS.2020.3011028.

[2] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.

[3] Networks, A. N. (2023, August 31). *What are the advantages and disadvantages of using pre-trained CNNS?*. Pre-trained CNNs: Benefits and Challenges for Image Tasks. https://www.linkedin.com/advice/0/what-advantages-disadvantages-using-7050897204365991936

[4] K, GOPALA KRISHNAN, Face Anti-Spoofing Using Deep Learning. Available at SSRN: https://ssrn.com/abstract=4089543 or http://dx.doi.org/10.2139/ssrn.4089543

[5] Anthony, Peter & Ay, Betul & Aydin, Galip. (2021). A Review of Face Anti-spoofing Methods for Face Recognition Systems. 1-9. 10.1109/INISTA52262.2021.9548404.

[6] “The Advantages and Disadvantages of a Mobile App.” *Objectiveit*, Objective, 2021, objectiveit.com/blog/the-advantages-and-disadvantages-of-a-mobile-app/. Accessed 28 Sept. 2023.

[7] Roomi, Mishal. “5 Advantages and Disadvantages of HTTP: Drawbacks & BENEFITS OF HTTP.” *HitechWhizz*, HitechWizz, 14 Aug. 2020, www.hitechwhizz.com/2020/08/5-advantages-and-disadvantages-drawbacks-benefits-of-http.html.

[8] Rolling, Mitch. “Why Transitioning to Renewable Energy Leads to Power Outages.” *American Experiment*, 26 Aug. 2020, www.americanexperiment.org/why-transitioning-to-renewable-energy-leads-to-power-outages/.

[9] “The Pros and Cons of Power Backup Systems.” *Energy5*, 22 Sept. 2023, energy5.com/the-pros-and-cons-of-power-backup-systems#anchor-3.

[10] DeCapua, Todd. “Front-End vs Back-End Performance Metrics for Mobile Apps.” *TechBeacon*, TechBeacon, 6 Feb. 2020, techbeacon.com/app-dev-testing/understanding-front-end-vs-back-end-performance-metrics-mobile-apps.

[11] GeeksforGeeks. (2023, January 3). *Determine the face tilt using OpenCV - Python*. GeeksforGeeks. https://www.geeksforgeeks.org/determine-the-face-tilt-using-opencv-python/

[12] GitHub. (n.d.). https://github.com/opencv/opencv/tree/master/data/haarcascades

[13] Tranter, Jeff. “Control Raspberry Pi GPIO Pins from Python.” *ICS - Integrated Computer Solutions*, ICS, 31 July 2019, www.ics.com/blog/control-raspberry-pi-gpio-pins-python. Accessed 28 Sept. 2023.

[14] Cassidy, Lance. “Arduino vs Raspberry Pi: Which Is the Best Board for You.” *Www.flux.ai*, Flux, 7 Feb. 2023, www.flux.ai/p/blog/arduino-vs-raspberry-pi-comparison. Accessed 25 Sept. 2023.

[15] A. S. Savanth, K. G. R. Manish, P. Narayan, M. L. Nikhil and V. G. Gokul, "Face Recognition System with 2D Anti-Spoofing," 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 2022, pp. 226-230, doi: 10.1109/AIC55036.2022.9848909.

[16] Breuss, Martin. “Python-Driven Web Applications – Real Python.” *Realpython.com*, Real Python, 1 Feb. 2021, realpython.com/python-web-applications/. Accessed 28 Sept. 2023.

[17] McKenzie, Cameron. “How to Install Apache’s Web Server on Windows 10 Quickly.” *Www.theserverside.com*, TechTarget, 15 Jan. 2022, www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Install-Apache-Web-Server-24-Windows-10-ServerRoot-Error. Accessed 28 Sept. 2023.