UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN
CS 437 - INTERNET OF THINGS

RAVN Shared-Smart-Lock System.

Rishi Gulati   Ansh Billimoria   Vatsal Chheda   Niranjana Balasubramanian

1 Deliverables

	Resource URL
GitHub	`https://github.com/vatsalchheda/CS437-Final-Project/tree/master/`
Video	`https://drive.google.com/file/d/1uieMvb7Fkuf0XqNCfmtiV6-6hm2EIE3e/view?usp=share_link`

2 Motivation

2.1 The Problem

Traditional safes, while providing a certain level of security, suffer from several limitations that make them less suitable for modern households. These issues include:

- *Limited access control*: Traditional safes typically use keys or combination locks, which limit the number of people who can access the safe. This can be problematic for households with multiple members who need access to the safe's contents or in situations where temporary access must be granted.
- *Physical key management*: The use of physical keys in traditional safes introduces the risk of losing or misplacing keys, which can lead to security breaches or the inability to access the safe's contents when needed.
- *Difficulty in tracking*: With traditional safes, it is tough to monitor who has accessed the safe and when, making it hard to identify security breaches or unauthorized access.
- *Limited integration*: Traditional safes lack integration with other smart devices or security systems, which can limit their functionality in a modern, connected household.

- *Lack of individual accountability*: In situations where multiple people share a traditional safe, it can be challenging to determine who last accessed the safe and what items were removed. This can lead to a lack of individual accountability and trust issues among users, especially if valuable items go missing or are misplaced.

2.2 The Existing Solutions

Current smart locks available in the market address several issues faced by traditional safes. We analyzed many existing smart lock applications using innovations like IoT, Blockchain, and facial recognition to mitigate the above problems [5, 13, 6, 3, 12]. A few key benefits of such smart locks include:

- *Enhanced access control*: Smart locks offer digital access control methods, such as smartphone apps, RFID cards, or biometric authentication, which allow for the creation of multiple access codes or keys. This makes it easier to grant and revoke access to various users as needed.
- *Elimination of physical key management*: By using digital access methods, smart locks eliminate the need for physical keys, thereby reducing the risk of lost or misplaced keys and enhancing overall security.
- *Integration with smart home systems*: Many smart locks can be integrated with existing smart home systems and other connected devices, allowing for centralized control and monitoring of the lock. This enables users to remotely access the lock, receive notifications about its status, and even automate specific actions based on predefined triggers.
- *Enhanced security features*: Smart locks often come with additional security features such as tamper detection, automatic locking and unlocking, and activity logs that record when the lock is accessed and by whom. These features help to increase the overall security of the lock and provide users with greater peace of mind.

2.3 The Gaps in Market

While smart locks have become increasingly popular in recent years, they still suffer from several limitations and challenges that have yet to be fully addressed by the market. Analyzing existing literature on security threats with IoT, and challenges with smart locks [7, 14, 10] we find concerns about security, difficulty in setup and use, access problems with multiple users, vulnerability to cyber attacks, reliability, connectivity issues, and high costs. A few of these gaps are listed below,

- *Lack of security*: Some smart locks may be vulnerable to hacking or other forms of digital attacks, compromising the security of the safe. Additionally, smart locks can still be susceptible to physical attacks, such as drilling or prying.

- *Difficulty of setup and use*: Setting up a smart lock may require technical knowledge and expertise, making it difficult for some users to install and configure the lock. Additionally, users may face challenges in using the lock's features or navigating its interface.
- *Cost*: High-quality smart locks can be expensive, which may deter some consumers from adopting the technology. The cost of maintaining and upgrading the lock, as well as any subscription fees for cloud-based services or additional features, can also be a financial burden for users.
- *Limited Accountability*: While smart locks can provide enhanced access control features such as multiple access codes or keys, they may not be able to identify who exactly accessed the lock at a given time. To address this issue, some smart locks may offer additional features such as individual user accounts or activity logs that track each user's access to the lock. However, even with this feature in select locks, there is still no provision for one authorized user to open the lock without the consent of other users.

As a result, there is a growing need for a more advanced, secure, and user-friendly solution that can address these gaps in the market.

2.4 Introducing RAVN

The RAVN Smart-Shared Lock System offers a cutting-edge solution that addresses the gaps in the market and surpasses the limitations of both traditional safes and existing smart locks. This system is designed to provide enhanced security, ease of use, and affordability, while also offering unique features to cater to the needs of modern households.

- *Four Levels of Authentication*: RAVN provides a comprehensive and robust security solution with four levels of authentication: password, facial recognition, voice authorization, and biometric fingerprint scanning. This multi-factor authentication approach ensures that only authorized users can access the safe and significantly reduces the risk of unauthorized access or security breaches.
- *Raspberry Pi-based System*: By utilizing Raspberry Pi technology, RAVN can be introduced at a lower price point than most commercial systems, making it a more affordable option for consumers. The Raspberry Pi platform also offers greater customization and adaptability, allowing for future enhancements and updates to the system.
- *Offline Authentication*: RAVN does not rely on internet access or cloud-based services for authentication, reducing the risk of security breaches due to hacking or connectivity issues. This also ensures that the system remains operational even during power outages or internet disruptions, providing users with a reliable and secure solution.
- *Joint Access Control*: This is the **USP** of our RAVN system. Similar to bank joint accounts, RAVN allows for the creation of shared accounts with n number of users, requiring all n users to simultaneously authenticate using the biometric sensor before granting access. This feature prevents unauthorized access by requiring the consent of all users, ensuring the contents of the safe remain secure and accessible only to the designated individuals.

- *Easy Setup and User Registration*: RAVN offers a simple and user-friendly setup process, enabling users to easily install and configure the lock without the need for technical expertise. The system also allows for effortless registration of new users, making it convenient for households with multiple members or situations where temporary access must be granted.
- *Communication and Activity Monitoring*: RAVN communicates with registered users via email, generating activity logs and alerting users to unauthorized access attempts. The system also captures fingerprint images and facial images of individuals attempting to access the lock without authorization, providing valuable information to users and legal authorities and enhancing the overall security of the system.

In conclusion, the RAVN Smart-Shared Lock System offers a superior solution that addresses the limitations of traditional safes and existing smart locks, providing advanced security, ease of use, and affordability. Its unique features, such as multi-factor authentication, joint access control, offline authentication, and user-friendly price-point make it an ideal choice for modern households seeking a reliable and secure locking solution.

3 Technical approach

The following section will provide a high-level overview of the system details, technical considerations, and requirements of our RAVN system.

3.1 Architecture Overview

The RAVN Smart-Shared Lock System is built using a Raspberry Pi as the central processing unit, responsible for managing user data, handling authentication, and controlling the locking mechanism. The system is composed of several key components, including a biometric fingerprint sensor, facial recognition web camera, a microphone for voice authorization, and a keypad for password input. The system setup also includes three LEDs, each of which lights up for every successful level of security clearance. The entire code base of the system will come stored in Pi and is designed with a modular and scalable architecture, ensuring seamless integration with future enhancements.

3.2 System Choices

The choice of the Raspberry Pi as the CPU provides the RAVN system with a powerful, cost-effective, and energy-efficient platform, making it suitable for a wide range of applications. Python is used as the primary programming language for Raspberry Pi. The Raspberry Pi-Python combination offers a large community of developers and extensive libraries, enabling the system to leverage existing resources and streamline development efforts.

There were a lot of different security-related sensors available in the market that gauge different parameters to secure identity and authority. Our choice of sensors was based on the following factors,

- *Small Size*: The sensors needed to be small and portable to allow for easy integration into the RAVN system and enable mobility and flexibility in deployment.
- *Low Power Consumption*: Low power consumption was a crucial factor in sensor selection to ensure energy efficiency and minimize the system's overall power requirements.
- *Low Hardware Cost*: The sensors had to be cost-effective to keep the overall system cost within budget and make it accessible to a wider range of users.
- *Reliability and Accuracy*: The sensors needed to be reliable and accurate in capturing and processing biometric data to ensure the integrity and security of the authentication process.
- *Run Time Consideration*: The sensors' run time, or the time taken to capture and process data, had to be fast enough to provide a seamless user experience and minimize delays in real-time.
- *Available Open Source Documentations*: Open-source documentation and resources were preferred to facilitate development and reduce time-to-market by leveraging existing solutions and knowledge.

Based on the above criteria we zeroed in on the biometric fingerprint sensor, facial recognition web camera, microphone, and a keypad were selected. These components work together to provide multi-factor authentication, ensuring a high level of security while maintaining user convenience.

3.3 Network Considerations

To ensure the security and privacy of user data, the RAVN system does not rely on internet access or cloud-based services or thrid-party APIs for authentication. Instead, all data processing and authentication are performed locally on the Raspberry Pi, minimizing the risk of data breaches and connectivity issues. This choice also enables the system to operate independently of external networks, ensuring consistent performance even during internet disruptions.

However, the RAVN system can be optionally configured to connect to a local network for sending email notifications and activity logs to registered users. This feature is implemented using a secure (HTTPS) and encrypted connection to protect user data and maintain privacy.

Furthermore, to ensure the security and confidentiality of user passwords, the RAVN system uses a one-way hashing algorithm (SHA256) to store them in the database. This means that once a password is hashed, it cannot be reversed to its original form, making it difficult for hackers to obtain the actual password even if they gain access to the system's database or source code.

Additionally, the RAVN system does not use salt in its hashing process. This is a conscious network consideration made by us. While using salted hashes is generally considered the best practice for password storage, in the context of the RAVN Smart-Shared Lock System, the use of unsalted SHA-256 hashes can be justified. This approach simplifies the password storage and verification process, improves performance, and reduces potential points of failure. Additionally, the RAVN system operates offline and implements multi-factor authentication, reducing the attack surface and making it more difficult for attackers to gain unauthorized access. The system also has a limited number of users, reducing the likelihood of hash collisions.

3.4 Components Integration and Circuit Diagram

As stated and justified in Section 3.2, we use the following components at the core of our RAVN security system,

A. Raspberry Pi 4

B. Logitech Web Camera

C. Standard Microphone

D. Adafruit Biometric Sensor

E. Generic LEDs

Figure 3.1 below gives the circuit diagram of our RAVN system after successfully integrating the core sensors.

3.5 Logic Flow

Figure 3.2 gives a high-level overview of the logic flow of the RAVN System.

3.6 Post Authorization Control Flow

After successfully completing the four-step authentication process, an authorized user has three options: open the vault, add another user, or open a joint account.

1. *Open the vault*: Upon selecting this option, the system prompts the user for final confirmation. After the user approves the action, the locking mechanism is activated, granting access to the contents of the safe. This additional confirmation step ensures that the user has a clear understanding of their action and minimizes the risk of accidental or unintended unlocking.

2. *Add another user*: In order to maintain security and control over access to the safe, only authorized users are allowed to add new users. When a user chooses this option, they must be accompanied by the new user. The new user is required to provide their name and email address for registration purposes. Following this, the RAVN system automatically runs a script (a detailed technical description will follow in Section 4 which prompts the new user to register their face, voice password, and biometric fingerprint.

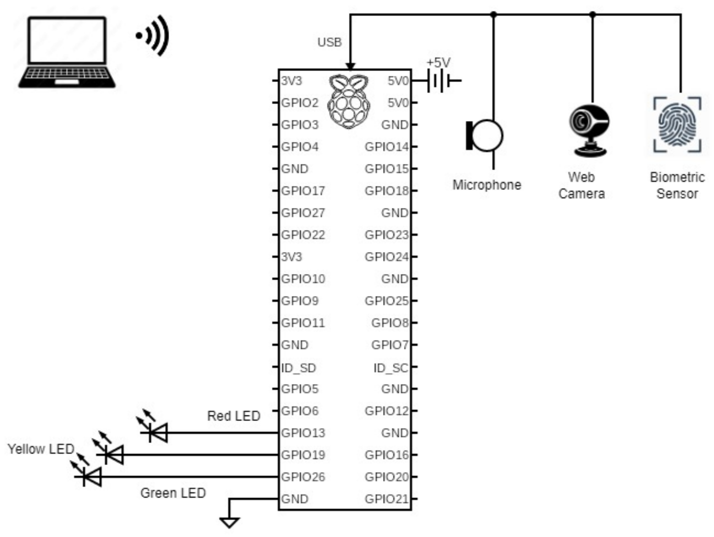


Figure 3.1: Circuit Diagram

3. *Access/Create a joint account*: If a user opts to open/create a joint account, they must provide the name of the secondary user (or users), who will then be prompted to authenticate their fingerprint. These users must be already registered with the system prior to opening or accessing a joint account. Upon successful fingerprint authentication, the locking mechanism is activated, and the lock is opened. This feature enables shared access to the safe while ensuring that all the owners consent to the action, providing an additional layer of security and accountability.

These features, combined with the robust multi-factor authentication process, make the RAVN Smart-Shared Lock System a versatile and secure solution for modern households, addressing the limitations of traditional safes and existing smart locks while offering a user-friendly and accessible experience.

4 Implementation Details

The RAVN Smart-Shared Lock System was developed using a combination of hardware modules, software packages, and algorithms, with various data structures and protocols in place to ensure efficient operation and security. In this section, we provide an overview of these implementation details.

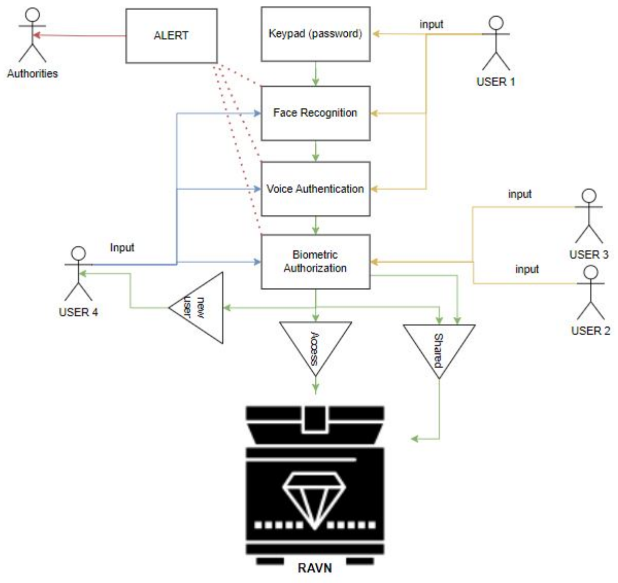


Figure 3.2: Logic Flow Diagram

4.1 Hardware Modules and Python Packages

Table 4.1 provides an unofficial link to resources that were referenced while creating the RAVN system.

	Resource URL
Face Recognition	`https://github.com/ageitgey/face_recognition`
Speech Recognition	`https://github.com/Uberi/speech_recognition`
Biometric sensor	`https://github.com/bastianraschke/pyfingerprint`
LEDs	`https://gpiozero.readthedocs.io/en/stable/`

4.2 Implementing Face Recognition

In comparison to [8], which uses only face recognition using a deep learning algorithm called Yolo, our system is designed for lighter use and contains two more layers of security ensuring that it doesn't require a heavy algorithm as described in the paper. Our system is similar to [16] which also uses the haar classifier method except that we also send an email in case of the arrival of an intruder. The authors in [2] propose using the LPBH algorithm to get clear images of people but we decide to avoid using it because it has a significant amount of disadvantages. LPBH algorithm tends to be sensitive to lighting and essentially fails to recognize faces if the lighting is poor. Additionally, it is sensitive to image quality and causes erroneous results if there is noise in the image. LPBH is also not good at capturing facial expressions, making it difficult to handle in everyday scenarios.

We will be utilizing several Python packages, including OpenCV, face_recognition, and imutils, to train our Raspberry Pi for facial recognition. Our dataset will consist of images that we collect, which will be processed using a file called *headshots.py*. This file will capture images of the user when the spacebar is pressed, and the user can press the esc key to stop taking pictures. Once we have gathered our dataset, we will run the *train_model.py* file to analyze the images and create a mapping between names and faces in the *encodings.pickle* file. Finally, we will use the *facial_recognition* file to recognize faces in a live camera feed.

The algorithm used to determine faces is called Haar Cascade. This is a library that is widely used for face recognition in computer vision. The Haar Cascade algorithm is based on the principle of feature extraction from images. It uses a set of simple, rectangular Haar-like features to detect objects in an image. These features are calculated by summing the pixel intensities in adjacent rectangular regions of the image. Each feature provides information about the texture, edges, and contrast in a particular region of the image.

During training, the Haar Cascade algorithm learns to classify an object by combining the responses of many individual features. It uses a machine learning technique called Adaboost to select a subset of the most relevant features for object detection. The algorithm then combines these features into a cascade of classifiers, each of which makes a decision about whether a region of the image contains a face or not. The Haar Cascade algorithm is computationally efficient and can process images in real-time.

If we recognize an authorized user we add the name in the logs, turn on the status indicator and the authorization can proceed to the next step. If an unrecognized user has been detected then a security email is sent to all the authorized users with a record of the log file and the image of the unknown person and the RAVN system locks itself.

4.3 Implementing Speech Recognition

In our system, we have used a text-to-speech conversion technique that captures a user's voice and converts it to text. The system proposed by [15] builds an IoT system that can recognize words spoken in both English and French. Our system focuses only on words spoken in English. We have used speech_recognition library that easily integrates with other Python libraries with cross-platform compatibility across multiple platforms, and easy-to-use API. To come to the decision of choosing this library, we went through [9] which explains the background of how speech recognition works. We also looked up [1] which builds a smart lock system based on speech recognition alone.

We utilized the *SpeechRecognition* Python package for our speech recognition module and integrated Google's Cloud Speech API to convert speech to text. To register a new user, they enter a passcode which is then hashed and stored in the system. When the user passes facial recognition, they are prompted to speak the passcode, which is then passed to the Google Cloud API for text conversion. The resulting text is hashed and compared to the saved password, and if it's a match, the next status LED is turned on and the authentication proceeds to the next step.

4.4 Implementing Biometric Sensing

From [11], we learned about the characteristics of fingerprints and how to use them to distinguish between people's fingerprints. Our system uses a fingerprint scanner that captures data from the user's fingerprint and sends it to the system to compare it with existing fingerprints recorded. [4] provides a detailed view of fingerprint scanners and the sensors used to build them.

We use the *pyfingerprint* library for fingerprint matching that provides simple and easy-to-use APIs for capturing and processing fingerprint images. This stage prompts the user to scan their biometric fingerprint on the device attached to the system. The device scans for the user's fingerprint to find a match.

When the user gets their fingerprint scanned, the image of the fingerprint is compared to the ones stored, and the accuracy score for it is calculated. If there is no match found, a message is printed to the user by the system before exiting. Additionally, if the network is configured, the system will send an email to the recognized users with the fingerprint of the intruder. This can be later used to trace and find the intruder. If there is a successful match, the third LED lights up, completing the authentication process.

4.5 Configuring Email Reporting

The RAVN Smart-Shared Lock System is designed to send email reports to registered users, providing them with crucial information about the system's activity. This feature is particularly useful in cases of unauthorized access attempts or when a new user is added to the system. The email notifications are generated and sent using Python's smtplib and email libraries.

The send_mail() function in the code snippet provided has two modes:

1. Mode 1: This mode is triggered when the system detects a potential security breach or unauthorized access attempt. It sends an urgent email notification to all registered users, containing an attached activity log and an image of the unauthorized access attempt. The email subject is marked as "[URGENT] RAVN Security System – Activity information.", and the body provides a brief description of the situation, advising users to consider calling emergency services.

2. Mode 2: This mode is activated when an authorized user adds a new user to the system. It sends an email update to all registered users, informing them about the new addition to the system. The email subject is marked as "[UPDATE] RAVN Security System – Activity Update.", and the body includes the names of the existing user and the new user, along with an attached activity log for reference.

To securely send emails, the function establishes a connection with the Gmail SMTP server using SSL encryption on port 465. In both modes, a MIMEMultipart object is created to compose the email. The email body is attached as plain text using the MIMEText class. For attachments, such as the activity log and image, the MIMEBase and MIMEImage classes are used, respectively. The attachment files are read, encoded using base64 encoding, and added to the MIMEMultipart object with appropriate headers. A success message is printed to the console upon successful completion of the email-sending process.

4.6 Data Flow

Figure 4.1 gives a high-level overview of the data flow of the RAVN System.

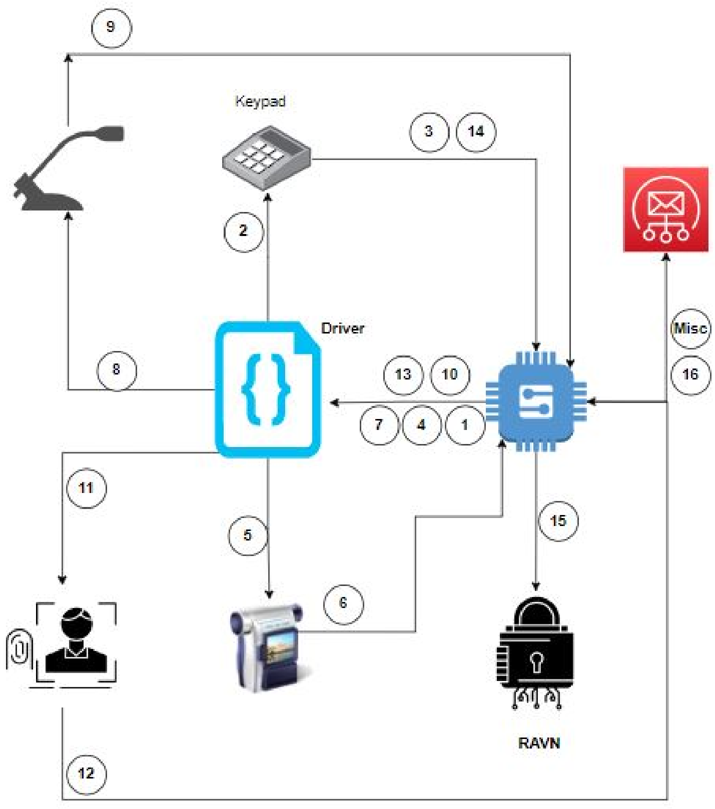


Figure 4.1: Data Flow Diagram

5 Results

The RAVN Smart-Shared Lock System successfully addresses the limitations and challenges faced by traditional safes and existing smart locks. By implementing a four-step authentication process, the system enhances security while providing a user-friendly experience. The results of the implementation and testing indicate that the system performs effectively in terms of security, ease of use, and accessibility.

Note: We highly encourage looking at the demonstration video linked in the Deliverables section of the document to get a holistic overview of the results of the implemented RAVN System.

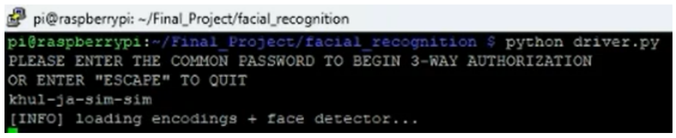
5.1 Keypad Password Results

The RAVN Smart-Shared Lock System incorporates an initial pincode/password authentication step that adds an additional layer of security to the system. Before users can proceed with the subsequent authentication steps (face recognition, voice authorization, and biometric fingerprint), they must first enter a valid pincode or password on the console.

This pincode/password authentication step provides several benefits:

- *Enhanced Security*: The initial pincode/password requirement adds another obstacle for potential intruders, making unauthorized access even more difficult.
- *Protection Against Brute Force Attacks*: By requiring a pincode/password input, the system effectively defends against brute force attacks, as the intruder must first crack this code before attempting to bypass the other authentication layers.
- *Controlled Access*: The pincode/password requirement ensures that only users with the correct credentials can proceed with the authentication process, providing a controlled access point to the system.

Figure 5.1 gives an overview of the initial password protection layer of security.



```
pi@raspberrypi: ~/Final_Project/facial_recognition
pi@raspberrypi:~/Final_Project/facial_recognition $ python driver.py
PLEASE ENTER THE COMMON PASSWORD TO BEGIN 3-WAY AUTHORIZATION
OR ENTER "ESCAPE" TO QUIT
khul-ja-sim-sim
[INFO] loading encodings + face detector...
```

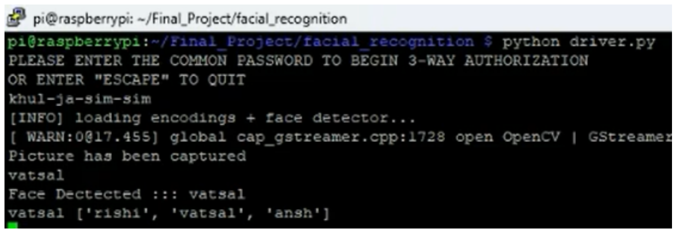
Figure 5.1: Keypad Password and Pincode Entry

The challenges with this stage involved storing passwords securely. It is critical for maintaining the overall security of the system. One approach we use to secure password storage is to store hashed versions of the passwords instead of plaintext. This way, even if an intruder gains access to the password storage, they would still need to crack the hash to retrieve the original password. For this, we choose a secure and reliable hashing algorithm. Inadequate hashing algorithms may be vulnerable to attacks or may not sufficiently protect the password data. In the context of the RAVN system, the use of the SHA-256 hashing algorithm provides a good balance between security and performance.

5.2 Face Recognition Results

For face recognition, our system accurately captures and detects the faces of people, ensuring no intruder can access the system. Among a vast variety of algorithms and libraries that can be used for face recognition, we chose Haar Cascade because, firstly, it is a fast and efficient algorithm that can detect objects in real-time, making it suitable for applications such as face detection in video streams. Second, it is highly accurate, with a low rate of false positives and false negatives. This is because the algorithm uses a sophisticated machine-learning technique called boosting, which combines multiple weak classifiers to form a strong classifier. Finally, the Haar cascade algorithm is flexible and can be trained to detect a wide range of objects, making it suitable for a variety of applications such as object recognition, face detection, and pedestrian detection.

The face recognition process is initiated and controlled by our script driver.py, which manages the data between Pi and the hardware sensors. Figure 5.2 gives an example of a case when one of the registered users, Vatsal, was successfully detected and recognized by the system.



pi@raspberrypi: ~/Final_Project/facial_recognition

```
pi@raspberrypi:~/Final_Project/facial_recognition $ python driver.py
PLEASE ENTER THE COMMON PASSWORD TO BEGIN 3-WAY AUTHORIZATION
OR ENTER "ESCAPE" TO QUIT
khul-ja-sim-sim
[INFO] loading encodings + face detector...
[ WARN:0@17.455] global cap_gstreamer.cpp:1728 open OpenCV | GStreamer
Picture has been captured
vatsal
Face Dectected ::: vatsal
vatsal ['rishi', 'vatsal', 'ansh']
```

Figure 5.2: Face Recognition Example

Implementing face recognition as a part of the RAVN Smart-Shared Lock System's authentication process presented several challenges. These challenges arose from the need to balance the system's performance, accuracy, and user experience. Some of the key challenges faced during face recognition implementation include:

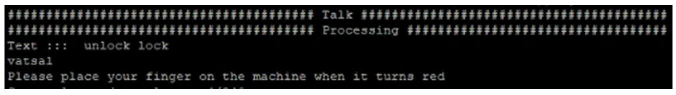
- *Frame Rate*: The frame rate at which the camera captures images can significantly impact the system's performance and user experience. A higher frame rate may improve the responsiveness and accuracy of the face recognition process but could also increase the computational load on the system, potentially leading to slower processing times and increased power consumption. Balancing frame rate with system performance was a critical challenge during implementation.

- *Accuracy*: Ensuring accurate face recognition is essential for maintaining the security of the system. However, achieving high accuracy can be challenging due to variations in lighting conditions, camera angles, and facial expressions. Experimenting with different face recognition algorithms, preprocessing techniques, and fine-tuning the model's parameters was necessary to improve accuracy while maintaining acceptable performance levels.

- *Tolerance of Prediction*: The tolerance level in face recognition determines the threshold for accepting or rejecting a face as a match. Setting this threshold too low may result in a high rate of false negatives, where authorized users are denied access. On the other hand, setting it too high may lead to false positives, where unauthorized users are granted access. Finding the optimal tolerance level required extensive testing and experimentation to strike the right balance between security and user experience.

5.3 Voice Authorization Results

For speech recognition, we use the speech_recognition library of Python. This library makes use of various algorithms and technologies to perform speech recognition, including audio recording, pre-processing, segmentation, feature extraction, speech recognition engines, and language models. It accurately captures the words spoken by the user and converts them to text. Figure 5.3 gives an overview of the speech-to-text conversion and hashing that goes on at the back-end during voice authentication.



```
######################################## Talk ########################################
######################################## Processing ##################################
Text :::  unlock lock
vatsal
Please place your finger on the machine when it turns red
```

Figure 5.3: Voice Authentication Example

Incorporating speech recognition as an authentication method in the RAVN Smart-Shared Lock System introduced several challenges. These challenges stem from the need to maintain a balance between the system's performance, accuracy, and user experience. Some of the key challenges faced during speech recognition implementation include:

- *Ambient Noise*: One of the primary challenges in speech recognition is dealing with ambient noise. Background noises, such as music, traffic, or conversations, can interfere with the system's ability to accurately recognize the user's voice. To mitigate this issue, noise reduction techniques and preprocessing methods were employed to improve the system's ability to distinguish the user's voice from the background noise.

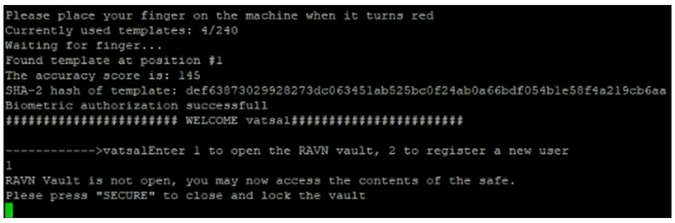
- *Accuracy*: Ensuring accurate speech recognition is crucial for maintaining the security of the system. However, achieving high accuracy can be challenging due to factors such as variations in speech patterns, accents, and pronunciations among users. Experimenting with different speech recognition algorithms, preprocessing techniques, and fine-tuning the model's parameters was necessary to improve accuracy while maintaining acceptable performance levels.

- *Real-time Processing*: For a seamless user experience, the speech recognition process should ideally work in real-time, providing immediate feedback to the user. However, achieving real-time processing can be challenging due to the computational requirements of speech recognition algorithms. Optimizing the system for real-time processing without compromising accuracy or security was a key challenge during implementation.

5.4 Fingerprint Authentication Results

The scanner captures the user's fingerprint and proceeds to analyze it for a match. Upon scanning the user's fingerprint, the system compares the captured image with the stored fingerprint images of authorized users. It then calculates an accuracy to determine the level of similarity between the scanned fingerprint and the stored ones. By performing this comparison and evaluating the accuracy score, the RAVN Smart-Shared Lock System is able to verify the user's identity through their unique fingerprint, ensuring an additional layer of security during the authentication process. The results of this process are illustrated in Figure 5.4.

Implementing fingerprint recognition as an authentication method in the RAVN Smart-Shared Lock System presented various challenges. These challenges are related to ensuring



```
Please place your finger on the machine when it turns red
Currently used templates: 4/240
Waiting for finger...
Found template at position #1
The accuracy score is: 145
SHA-2 hash of template: def63873029928273dc063451ab525bc0f24ab0a66bdf054b1e58f4a219cb6aa
Biometric authorization successfull
######################## WELCOME vatsal########################

------------>vatsalEnter 1 to open the RAVN vault, 2 to register a new user
1
RAVN Vault is not open, you may now access the contents of the safe.
Plese press "SECURE" to close and lock the vault
```

Figure 5.4: Biometric Authorization Example

the system's performance, accuracy, and user experience. Some of the key challenges faced during fingerprint recognition implementation include:

- *Unclean Sensor*: A fingerprint sensor's performance may be negatively impacted by dirt, grease, or other contaminants present on its surface. An unclean sensor can lead to inaccurate or incomplete fingerprint scans, resulting in false negatives or difficulty in recognizing authorized users. Regular maintenance and cleaning of the sensor, as well as encouraging users to clean their fingers before scanning, can help mitigate this issue.
- *Misaligned Finger Placement*: Accurate fingerprint recognition relies on proper finger placement on the sensor. Misaligned or partial finger placement can lead to incomplete or distorted scans, reducing the system's ability to recognize the user.
- *False Positives and False Negatives*: Balancing the trade-off between false positives (unauthorized users granted access) and false negatives (authorized users denied access) is a significant challenge in fingerprint recognition systems. Tuning the system's parameters and thresholds, as well as employing advanced matching algorithms, helped us achieve an optimal balance between security and user experience.

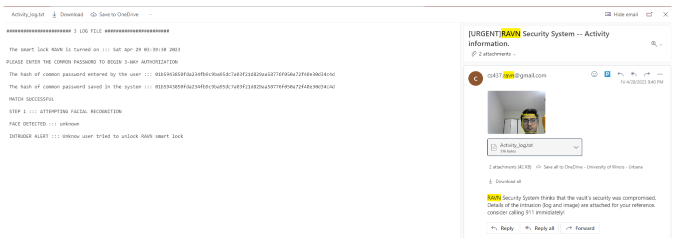
5.5 Email Reporting Results

In the RAVN Smart-Shared Lock System, email reporting plays a crucial role in keeping users informed and aware of the activities and security events related to their smart lock. This feature ensures that users receive timely notifications about potential security breaches, unauthorized access attempts, as well as updates on the addition of new authorized users. By leveraging email reporting, RAVN aims to provide users with greater transparency and control over their smart lock system, fostering a sense of trust and confidence in the overall security and performance of the solution.

Figure 5.6 gives an example of a reported email with reporting mode 2, and 5.5 gives an example of a reported email when the RAVN system thinks that it has detected an intruder.

6 Future Scope and Limitations

One of the primary constraints of the current smart lock system is the need for the user to execute the authentication script, which can be arduous. In addition, the facial recognition



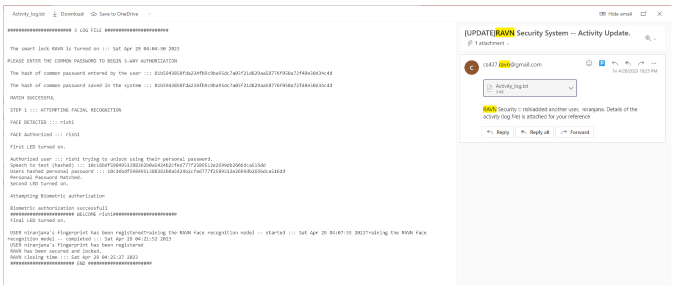
```
######################## 3 LOG FILE ########################


 The smart lock RAVN is turned on ::: Sat Apr 29 03:39:30 2023

PLEASE ENTER THE COMMON PASSWORD TO BEGIN 3-WAY AUTHORIZATION

 The hash of common password entered by the user ::: 01b5943850fda234fb9c9ba95dc7a03f21d829aa58776f050a72f40e30d34c4d

 The hash of common password saved in the system ::: 01b5943850fda234fb9c9ba95dc7a03f21d829aa58776f050a72f40e30d34c4d

 MATCH SUCCESSFUL

 STEP 1 ::: ATTEMPTING FACIAL RECOGNITION

 FACE DETECTED ::: unknown

 INTRUDER ALERT ::: Unknow user tried to unlock RAVN smart lock
```

[URGENT]RAVN Security System -- Activity information.

cs437.ravn@gmail.com
Fri 4/28/2023 9:40 PM

RAVN Security System thinks that the vault's security was compromised. Details of the intrusion (log and image) are attached for your reference. consider calling 911 immidiately!

Figure 5.5: Reported Email, mode 1



```
######################## 3 LOG FILE ########################


 The smart lock RAVN is turned on ::: Sat Apr 29 04:04:50 2023

PLEASE ENTER THE COMMON PASSWORD TO BEGIN 3-WAY AUTHORIZATION

 The hash of common password entered by the user ::: 01b5943850fda234fb9c9ba95dc7a03f21d829aa58776f050a72f40e30d34c4d

 The hash of common password saved in the system ::: 01b5943850fda234fb9c9ba95dc7a03f21d829aa58776f050a72f40e30d34c4d

 MATCH SUCCESSFUL

 STEP 1 ::: ATTEMPTING FACIAL RECOGNITION

 FACE DETECTED ::: rishi

 FACE Authorized ::: rishi

 First LED turned on.

 Authorized user ::: rishi trying to unlock using their personal password.
 Speech to text (hashed) ::: 10c16bdf5984951388362b0a5424b2cfed777f2589512e2699db2666dca516dd
 Users hashed personal password ::: 10c16bdf5984951388362b0a5424b2cfed777f2589512e2699db2666dca516dd
 Personal Password Matched.
 Second LED turned on.

 Attempting Biometric authorization

 Biometric authorization successfull
 ######################## WELCOME rishi########################
 Final LED turned on.

 USER niranjana's fingerprint has been registeredTraining the RAVN face recognition model -- started ::: Sat Apr 29 04:07:55 2023Training the RAVN face
recognition model -- completed ::: Sat Apr 29 04:21:52 2023
 USER niranjana's fingerprint has been registered
 RAVN has been secured and locked.
 RAVN closing time ::: Sat Apr 29 04:25:27 2023
 ######################## END ########################
```

[UPDATE]RAVN Security System -- Activity Update.

cs437.ravn@gmail.com
Fri 4/28/2023 10:25 PM

RAVN Security ::: rishiadded another user, niranjana. Details of the activity (log file) is attached for your reference

Figure 5.6: Reported Email, mode 2

technology employed by the system has limitations. The models are trained on a small subset of user images, causing the system to potentially identify an unauthorized user as an authorized one, or recognize someone else as an authorized user if they share similar characteristics such as a beard or glasses.

Another restriction arises when a user has multiple installed systems, and they must manually authorize themselves on each one, a cumbersome process that could be mitigated by incorporating an option to import data or users. Moreover, authorized users are unable to determine which lock is sending an alert email when an intrusion occurs or when adding a new user because there is no naming functionality for each device.

The system's effectiveness may be reduced in low-light situations or when the camera is not appropriately positioned, leading to false negative or false positive outcomes. Furthermore, the voice recognition feature may not function effectively for individuals with speech impairments or accents that the system has not been trained on.

The future scope of this smart lock system project is extensive. The first improvement could be integrating the system into different security systems like door locks, vaults, and lockers. This could include adding a button or motion sensors to start the authorization process automatically. Another potential improvement would be to add a screen to provide users with feedback on each security level cleared and ensure their face is in the frame during facial recognition.

Furthermore, cloud infrastructure could be implemented to allow users to connect to their lock and access the camera from anywhere. A centralized user management system could also be implemented, enabling users to be authorized once and having the authorization data synced across all systems. A naming functionality could also be added to each device, allowing authorized users to identify the device from which the warning or alert email is being received.

To improve the system's effectiveness in low-light situations or when the camera is not positioned correctly, infrared cameras or additional lighting could be added to the system. This will help in capturing clear images of the user's face, thereby improving the accuracy of the facial recognition system. Finally, the system could be trained on a diverse dataset that includes different accents and speech patterns. This will help improve the accuracy of the voice recognition system for a wider range of users.

References

[1] Shreya Aggarwal and Sachin Sharma. Voice based secured smart lock design for internet of medical things: An artificial intelligence approach. In *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, pages 1–7. IEEE, 2022.

[2] Navya Santhoshi Akkisetti. Door lock system based on face recognition using arduino. *Available at SSRN 4236516*, 2022.

[3] Tri-Nhut Do, Cong-Lap Le, and Minh-Son Nguyen. Iot-based security with facial recognition smart lock system. In *2021 15th International Conference on Advanced Computing and Applications (ACOMP)*, pages 181–185, 2021.

[4] Stanley Goodner. Wondering about fingerprint scanners? here's what you should know, Aug 2021.

[5] Muhammad Sabirin Hadis, Elyas Palantei, Amil Ahmad Ilham, and Akbar Hendra. Design of smart lock system for doors with special features using bluetooth technology. In *2018 International Conference on Information and Communications Technology (ICOIACT)*, pages 396–400, 2018.

[6] Donhee Han, Hongjin Kim, and Juwook Jang. Blockchain based smart door lock system. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1165–1167, 2017.

[7] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. Smart locks: Lessons for securing commodity internet of things devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, page 461–472, New York, NY, USA, 2016. Association for Computing Machinery.

[8] Minh-Khoi Le-Nguyen, Dinh-Thuan Le, Van-Hoa Nguyen, Long-Phuoc Tôn, Khuong Nguyen-An, et al. Hunting phishing websites using a hybrid fuzzy-semantic-visual approach. In *2021 15th International Conference on Advanced Computing and Applications (ACOMP)*, pages 38–45. IEEE, 2021.

[9] Ilias Papastratis. Speech recognition: A review of the different deep learning approaches, Jul 2021.

[10] Marko Pavelić, Zvonimir Lončarić, Marin Vuković, and Mario Kušek. Internet of things cyber security: Smart door lock system. In *2018 International Conference on Smart Systems and Technologies (SST)*, pages 227–232, 2018.

[11] Vivek R, Gokul D, Jai R, Mukilan Sriram, and Nihal V. A detailed review on fingerprint door lock system. *International Journal of Engineering Applied Sciences and Technology*, 6, 08 2021.

[12] J. Sa-ngiampak, C. Hirankanokkul, Y. Sunthornyotin, J. Mingmongkolmitr, S. Thunprateep, N. Rojsrikul, T. Tantipiwatanaskul, K. Techapichetvanich, A. Pongsawang, T. Prayoonkittikul, U. Wattanakulchart, N. Prompoon, C. Ratanamahatana, and M. Pipattanasomporn. Lockerswarm: An iot-based smart locker system with access sharing. In *2019 IEEE International Smart Cities Conference (ISC2)*, pages 587–592, 2019.

[13] M Shanthini, G Vidya, and R Arun. Iot enhanced smart door locking system. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 92–96, 2020.

[14] Maria Stoyanova, Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Pallis, and Evangelos K. Markakis. A survey on the internet of things (iot) forensics: Challenges, approaches, and open issues. *IEEE Communications Surveys Tutorials*, 22(2):1191–1221, 2020.

[15] Nagarjuna Telagam, U Somanaidu, M Arun Kumar, M Sabarimuthu, and Nehru Kandasamy. Iot based secure lock/unlock system using google assistant based english and french languages. *International Journal of Online & Biomedical Engineering*, 17(10), 2021.

[16] Sharvani Yedulapuram, Rajeshwarrao Arabelli, Kommabatla Mahender, and Chintoju Sidhardha. Automatic door lock system by face recognition. In *IOP Conference Series: Materials Science and Engineering*, volume 981, page 032036. IOP Publishing, 2020.