Project Proposal: Smart Biometric Access Control System for Shared Living Environments

ECE 445 / Senior Design Project

Jihao Li
Mujia Li
Shixuan Ma
Denghan Xiong

March 23, 2026

1 Introduction

1.1 Objective

The primary objective of this project is to develop an automated, high-security, and low-cost smart access control system tailored for shared living environments, such as university dormitories and shared apartments. Traditional access methods, such as mechanical keys or alphanumeric passwords, suffer from significant vulnerabilities: keys can be physically duplicated or lost, while passwords can be forgotten or shared illicitly. Our project aims to replace these outdated methods with a non-contact biometric solution utilizing a Raspberry Pi, computer vision, and facial recognition technology.

The system is designed to provide seamless entry by detecting a person's approach through a Passive Infrared (PIR) sensor, capturing high-definition images via a USB camera, and performing real-time facial feature extraction and comparison against a local database. Upon successful identification, a relay module is triggered to electronically unlock the door. Furthermore, the system incorporates a centralized logging mechanism, a graphical user interface (GUI), and an additional security enhancement module with liveness verification, voice-based secondary verification, and a coercion alert mechanism to improve both convenience and safety.

1.2 Background

In contemporary urban living, shared accommodation has become the norm for students and young professionals. Security in these environments is often fragmented. According to industry security reports, ”lost keys” represent one of the highest operational costs for dormitory management, often requiring complete lock replacements to maintain security integrity. Moreover, traditional locks provide no audit trail; there is no record of who entered a room or at what time.

Existing ”smart locks” on the market often rely on Bluetooth or Wi-Fi-connected apps, which require the user to pull out a smartphone—a process that can be as cumbersome as finding a key. High-end biometric systems exist but are often prohibitively expensive for large-scale dormitory deployment. Our solution leverages the affordability of the Raspberry Pi ecosystem and open-source computer vision libraries (OpenCV) to create a ”commercial-grade” biometric lock at a fraction of the cost, prioritizing local data processing to ensure resident privacy.

1.3 Benefits and Features

Benefits:

- **Enhanced Security:** Biometric data is significantly harder to forge than a physical key or a 4-digit PIN.
- **Audit Capability:** Every entry attempt (successful or failed) is logged with a timestamp and image, providing accountability.
- **Operational Efficiency:** Eliminates the need for physical key management and lock re-coring.
- **User Convenience:** True ”hands-free” entry for residents carrying groceries or luggage.
- **Liveness Verification:** The system includes a liveness check to reduce the risk of spoofing attacks using printed photos or replayed facial media.
- **Voice-Based Emergency Verification:** A microphone-based voice recognition subsystem can request a spoken passphrase during suspicious situations, providing an extra layer of identity verification and enabling a silent alarm trigger under coercion.

Features:

- **Dual-Stage Detection:** Uses PIR motion sensing to save power and only activate the camera when a person is present.
- **Real-Time Processing:** Multithreaded software architecture ensures identification occurs in under 1.5 seconds.
- **Robust GUI:** A PyQt5-based interface provides real-time feedback and system status to the user.
- **Privacy-Centric:** All facial recognition and database storage occur locally on the Raspberry Pi; no sensitive biometric data is uploaded to the cloud.

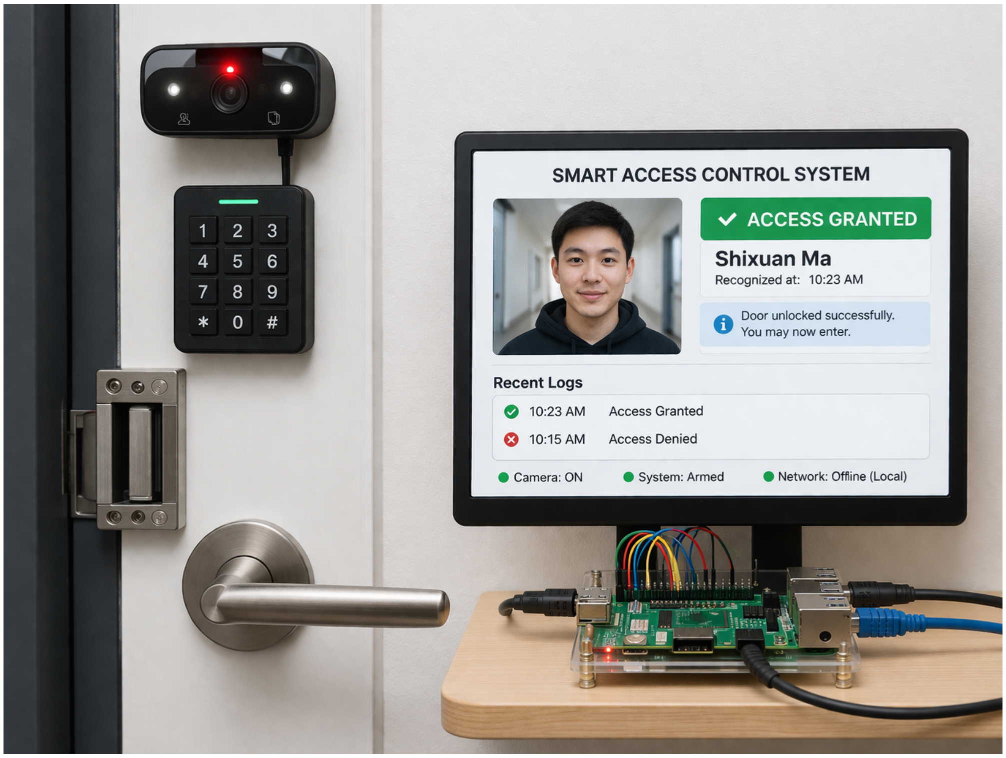
1.4 High-Level Requirements

To ensure the project is successful and verifiable, the following three high-level requirements must be met:

1. The system must detect a human approaching within a range of 2.0 meters and trigger the camera capture sequence within 500ms of detection.

2. The facial recognition algorithm must achieve a False Acceptance Rate (FAR) of less than 0.1% and a False Rejection Rate (FRR) of less than 5% under standard indoor lighting conditions (300-500 lux).

3. The total latency from initial motion detection to the relay activation (unlocking) must not exceed 2.0 seconds to maintain a positive user experience.



SMART ACCESS CONTROL SYSTEM

✓ ACCESS GRANTED

Shixuan Ma

Recognized at: 10:23 AM

Door unlocked successfully.
You may now enter.

Recent Logs

✓ 10:23 AM Access Granted

✗ 10:15 AM Access Denied

● Camera: ON ● System: Armed ● Network: Offline (Local)

Figure 1: Prototype implementation of the smart biometric access control system, including the door-side sensing unit, keypad, monitoring display, and Raspberry Pi control platform.

2 Design

2.1 Block Diagram

The system architecture is divided into three primary modules: the Hardware/Sensor Module, the Image Processing Subsystem, and the User Interface/Management Subsystem. A centralized Power Module provides regulated 5V DC to the Raspberry Pi and peripheral sensors.

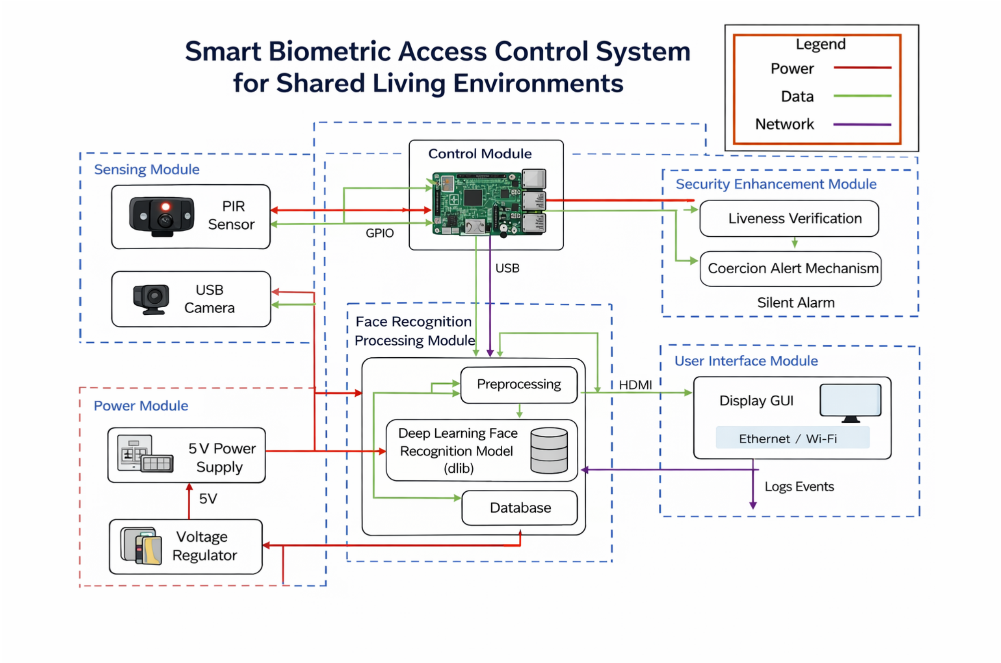


Figure 2: System block diagram of the smart biometric access control system for shared living environments.

2.2 Functional Overviews of Subsystems

2.2.1 Subsystem 1: Hardware and Sensing

This subsystem acts as the physical interface between the environment and the digital controller.

- **PIR Motion Sensor:** A digital infrared sensor that monitors thermal changes. It acts as a system wake-up trigger to prevent the Raspberry Pi from constantly running CPU-intensive vision algorithms.

- **USB Camera Module:** Captures 1080p video frames. It provides the raw visual data required for the recognition engine.
- **Relay Control Module:** An opto-isolated 5V relay. When it receives a high signal from the RPi GPIO, it closes the circuit for an electronic strike or magnetic lock to physically open the door.

2.2.2 Subsystem 2: Image Processing and Recognition

The "brain" of the system, responsible for converting pixels into identity.

- **Preprocessing (OpenCV):** Raw frames are resized and converted to grayscale to reduce computational load. Histogram equalization is applied to normalize lighting.
- **Face Recognition Module:** Uses a Deep Learning (dlib/HOG) or LBPH (Local Binary Patterns Histograms) approach to extract a 128-dimension vector representing the face. This vector is then compared against known vectors in the database using Euclidean distance.

2.2.3 Subsystem 3: UI and System Management

Ensures the system is usable and maintainable.

- **PyQt5 GUI:** Displays the live camera feed and a visual "Access Granted/Denied" prompt.
- **Multithreading Framework:** Separates the GUI thread from the recognition thread to prevent the interface from freezing during heavy computation.
- **Local SQLite Database:** Stores user profiles (names and face encodings) and an encrypted log of all access events.

2.3 Requirements and Verifications (R&V Tables)

Requirement	**Verification**
1. PIR Sensor: Must output a 3.3V High signal when a human-sized heat source moves within 2m.	1. Position a test subject at 2m distance. Use a multimeter on the PIR signal pin to confirm voltage $\geq 3.0V$ upon motion.
2. Camera Latency: Camera must initialize and capture a frame within 400ms of a PIR trigger.	2. Use a software timestamp to log the PIR interrupt time and the first frame buffer fill time. Calculate the delta.
3. Relay Module: Must activate and maintain a closed circuit for exactly 5 seconds upon a "Grant" signal.	3. Trigger the relay and use an oscilloscope or stopwatch to verify the duration of the "on" state is $5s \pm 0.5s$.
4. Face Recog Accuracy: System must correctly identify authorized users in at least 19 out of 20 trials (95%) in standard lighting.	4. Run a batch test with 20 attempts from 5 different authorized users. Record successful matches.
5. Power Stability: The 5V supply must not drop below 4.75V when both the RPi CPU is at 100% and the Relay is active.	5. Use an oscilloscope to monitor the 5V rail while running a stress test script and triggering the relay.

Table 1: Requirements and Verification Table.

3 Tolerance Analysis

The most critical component posing a risk to the successful completion of this project is the **Facial Recognition Processing Latency**. If the processing time is too high, the system will fail to meet the "real-time" requirement, leading to user frustration.

Mathematical Analysis of Latency: The total latency (T_{total}) is defined by:

$$T_{total} = T_{detect} + T_{capture} + T_{preprocess} + T_{extraction} + T_{match}$$

On a Raspberry Pi 4B (1.5GHz Quad-core), the extraction phase using a HOG-based model is the bottleneck.

1. **Extraction ($T_{extraction}$):** A single 640x480 frame extraction takes approximately $800ms$ per face. 2. **Preprocessing ($T_{preprocess}$):** Resizing and Grayscale conversion takes $\approx 50ms$. 3. **Matching (T_{match}):** Comparing a 128-d vector against a database of 50 users takes $\approx 10ms$ using a KD-Tree.

Therefore, $T_{total} \approx 500ms(PIR/Camera) + 50ms + 800ms + 10ms = 1.36s$.

Constraint: To ensure the system meets the high-level requirement of $< 2.0s$, we must ensure the face detection area is limited. If the system detects multiple faces, the extraction time scales linearly: $T_{ext\_total} = n \times 800ms$. To manage this, our code will implement a "Central Face Only" filter. By ignoring background faces and only processing the face largest in the frame (the person standing directly in front of the door), we cap $n = 1$.

Verification via Rudimentary Simulation: We will perform a Monte Carlo simulation of lighting conditions. Lighting directly impacts $T_{preprocess}$. Under low light, noise increases, potentially increasing the time for the HOG filter to converge. We will test the system across 100-800 lux. If $T_{extraction}$ exceeds $1.4s$ in low light, we will implement a hardware LED fill-light triggered by the PIR sensor to normalize the environment.

4 Ethics and Safety

4.1 Ethical Guidelines (IEEE/ACM)

In accordance with the **IEEE Code of Ethics, Section 1.1**, we hold the safety, health, and welfare of the public as paramount. Our system handles sensitive biometric data, which introduces significant ethical responsibilities:

- **Privacy and Data Protection:** We follow the principle of "Privacy by Design." No images or face encodings are transmitted over the internet. The local database is encrypted using AES-256. If the device is stolen, the biometric data remains unrecoverable.
- **Bias in AI:** Facial recognition algorithms have historically shown bias against certain demographics. We will use a diverse training set (dlib's pre-trained model) and perform validation across different skin tones and genders to ensure equitable access for all residents.
- **Transparency:** Users will be informed that the system uses facial recognition through clear signage, complying with local surveillance and data collection laws.

4.2 Safety Concerns

The system involves both low-voltage electronics and mechanical door hardware.

- **Electrical Safety:** While the RPi operates at 5V, the relay might interface with 12V or 24V electronic locks. We will use opto-isolators to prevent high-voltage spikes from reaching the logic board. All wiring will be housed in a non-conductive, flame-retardant enclosure.
- **Emergency Fail-Safe:** In the event of a power failure, the electronic lock must revert to a "Fail-Secure" or "Fail-Safe" mode as dictated by local fire codes. We will include a physical manual override (a standard key cylinder) to ensure residents are never trapped inside or locked out during an emergency.
- **Physical Installation:** The camera and PIR sensor must be securely mounted to prevent injury from falling components.

5 Citations

1 Intel Corporation, ”PIR Sensor Datasheet,” 2023. [Online].

2 Raspberry Pi Foundation, ”Raspberry Pi 4 Model B Product Brief,” 2019.

3 IEEE, ”IEEE Standard for Ethically Aligned Design,” IEEE Std 7000-2021.

4 dlib Library, ”High-performance facial landmark detection,” [Online].

5 PyQt5 Documentation, ”Riverbank Computing,” 2024.