Smart Security Home System

Abstract

The Smart Security Home System is a cost-effective, AI-powered solution designed to enhance residential security and automation through real-time facial recognition and unified appliance control. Leveraging the YOLOv8-Face model, the system delivers high-accuracy face detection and identification while operating entirely offline, ensuring data privacy and eliminating reliance on cloud infrastructure. Its innovation lies in the integration of AI-based recognition with fallback authentication methods, system resilience features, and a centralized interface for managing both AI-enabled and IoT-compatible home appliances. Performance evaluation shows that YOLOv8 significantly outperforms traditional methods like Haar and HOG, achieving a 99.75% detection rate with an average frame processing time of just 15.66 milliseconds. These results confirm the system's reliability, speed, and adaptability in real-world environments.

Introduction

Conventional door entry mechanisms such as mechanical locks and numeric codes are vulnerable to various security threats including unauthorized duplication, theft, and the sharing of access credentials. Biometric authentication offers a more secure alternative by leveraging inherent physiological traits, which are harder to replicate. However, traditional biometric systems often require specialized hardware like infrared cameras or cloud-based infrastructure, making them impractical for small-scale deployments or prototype development due to privacy risks and high costs (Zhao et al., 2003).

To overcome these challenges, the SmartSecureHome project focuses exclusively on a laptop-hosted facial recognition sub-system. It utilizes standard webcams and open-source embedding models to authenticate users locally, without reliance on external servers. This approach ensures that basic security requirements are met within a constrained resource environment, while also upholding user privacy (Schroff et al., 2015; Amos et al., 2016).

Problem Statement and Objectives

AI is enhancing security systems with facial recognition, object tracking, anomaly detection, and motion detection, improving accuracy and reducing false alarms. Consumers are increasingly adopting security solutions that integrate with Amazon Alexa, Google Assistant, and Apple Siri, allowing hands-free control and monitoring. Many companies now offer cloud-based security monitoring services, providing 24/7 surveillance, emergency response, and secure storage for recorded footage. However, while AI-powered security systems offer advanced features, consumer privacy concerns regarding data handling and potential misuse remain a challenge for market adoption.

With the increasing use of smart technologies like AI and IoT in everyday life, many residents in Singapore are looking for smarter and more secure ways to manage their homes. In HDB (Housing & Development Board) flats, where space is limited and cost is of a major consideration, traditional door entry methods such as keys and PIN codes are still widely used but they come with security risks like theft, keys/PIN duplication, or shared access. While biometric systems like facial recognition can offer better security, they often require expensive hardware or cloud-based services, which may not be practical or affordable for most HDB households. This creates a need for a simple, cost-effective, and privacy-friendly smart home security solution that works well in HDB environments without relying on complex setups or internet connectivity.

This project focuses on two main objectives. First, we aim to create a facial recognition door access system that could reliably identify authorized users with at least 80% accuracy in real-world conditions, removing the need for physical keys and enhancing home security. Second, to build a single, easy-to-use control platform that allows users to manage both door access and everyday appliances in one place. This platform shall support seamless integration of AI-enabled and non-AI appliances (IoT) through wireless connectivity.

Background

Face recognition systems have evolved significantly over the past decade, with many experiments exploring detection accuracy, speed, and robustness. One of the most influential works was FaceNet, developed by Schroff et al. (2015), which introduced a deep convolutional neural network trained with a triplet loss function to map faces into a 128-dimensional embedding space. Their experiments achieved over 99% verification accuracy on the Labeled Faces in the Wild (LFW) dataset. While this approach was highly accurate, it relied on separate face detection and face embedding steps, making the real-time implementation complex and computationally intensive—especially for edge devices or embedded systems.

To overcome these challenges, recent advances have focused on unified, single-stage detection and recognition models. The release of YOLOv8-Face by Ultralytics marked a notable improvement in this domain. This model combines high-speed face detection and lightweight embedding extraction in a single pipeline, capable of processing over 50 frames per second on GPU-enabled systems with competitive accuracy. These experiments also demonstrated the model's resilience under varied lighting conditions and partial occlusions. However, as a limitation, YOLOv8-Face still shows performance degradation on edge cases such as extreme facial angles or low-resolution video streams, which could impact its effectiveness in uncontrolled environments.

Parallel to advancements in facial recognition, smart security research has increasingly emphasized the importance of multi-factor authentication (MFA) in Internet of Things (IoT) systems. For instance, El-Hajj et al. (2019) conducted a comprehensive survey of IoT authentication schemes, highlighting how integrating fallback methods such as PIN codes or RFID with primary biometric authentication improves reliability while minimizing false acceptance rates. These hybrid models ensure users retain access even when biometric input fails due to lighting or angle issues. However, many existing implementations still lack broader system functionalities such as real-time logging, alert notifications, and fault tolerance—features essential for smart home systems deployed in dynamic real-world environments.

From these studies, several insights guided the development of our proposed AI-powered Smart Security Hub. First, we learned that unified models like YOLOv8-Face can significantly improve speed and deployment ease compared to two-stage pipelines like FaceNet. Second, we observed the necessity of fallback authentication to ensure continuous access when face recognition fails. Lastly, we recognized a research and implementation gap in building fault-tolerant, log-aware, and self-healing security systems.

Motivated by these findings, our system combines YOLOv8-Face for face recognition with PIN and RFID fallback authentication. It is also enhanced with features like simulated battery backup, automated self-healing routines, and tamper-evident logging to ensure operational continuity even under hardware or power failures. These design choices are driven directly by the strengths and limitations observed in prior experiments, and form the foundation of the AI techniques we describe in the next section.

AI Engineering Lifecycle

The AI engineering lifecycle for the Smart Security Home System unfolds across six iterative stages. It begins with Concept & Requirements Definition, during which stakeholder interviews and HDB‑specific use‑case analyses establish both functional needs—such as facial recognition, fallback authentication, and appliance control—and non‑functional targets, including at least 80 % recognition accuracy and an end-to-end latency below 200 ms. In the Data Preparation stage, diverse webcam footage captured under varied lighting and pose conditions is annotated with identity labels and augmented (via rotations and occlusions) to form a robust training corpus.

During Model Development & Validation, the YOLOv8-Face architecture is configured with a 640 × 640 input resolution and tuned confidence and IoU thresholds (0.35 and 0.45, respectively) before being trained in PyTorch; successive hyperparameter searches and edge-case evaluations yield inference speeds of 50 fps and recognition rates meeting the 80 % benchmark. The System Integration phase embeds this AI module within a Streamlit application, coupling it with administrative registration workflows, PIN-fallback logic, event-logging services, and appliance-control interfaces; continuous integration pipelines then automate unit and integration tests and deliver nightly prototype builds.

In Deployment & Monitoring, the integrated prototype operates on a standard laptop, continuously logging system uptime, fault counts, and authentication outcomes while surfacing real-time alerts and metrics through the UI. Finally, Maintenance & Continuous Improvement practices—including scheduled data-refresh cycles to retrain the model on newly collected face samples and failure logs, and containerized deployments—ensure that the system remains reliable, secure, and adaptable to evolving user needs and environmental conditions.

Requirements Specification

Functional Requirements

The Smart Security Home System must fulfill a collection of interrelated functional requirements that together ensure secure and reliable operation. First, the face-registration workflow requires an administrator to authenticate via a predefined password before the system captures up to thirty video frames through OpenCV and applies the YOLOv8-Face detection pipeline to each frame. A quality score—based on face bounding-box area and centring—is computed for every detected facial embedding, and the highest-scoring 128-dimensional vector is persisted in the local face_database.pkl. During authentication, the system acquires a live frame batch, extracts embeddings, and conducts a cosine-similarity comparison (tolerance 0.5) against stored records; a successful match immediately grants access, updates st.session_state.authenticated, records the user identity, and logs an “Access” event. If no match is found, a PIN-based fallback is invoked, prompting the user for a six-digit code that is validated against the FALLBACK_PIN constant. Correct PIN entry likewise grants access and logs success, whereas incorrect attempts log a “Security Alert” and generate a warning. All security events—including registrations, authentications, fallback attempts, and system faults—are timestamped and appended both to an in-memory session log and to the append-only CSV file access_log.csv via the log_event() function, while critical conditions trigger on-screen toasts through send_alert(). Following successful authentication, the system presents toggles for lighting, alarm arming, and Smart TV power within the Streamlit interface; each control operation respects battery-backup constraints and simulation mode, logs state changes, and updates real-time status displays. Finally, administrators may simulate power outages or camera faults, trigger self-healing, clear the face database (with password confirmation), and export the full access log via sidebar controls, ensuring comprehensive management of system operation and resilience.

Non-Functional Requirements

In addition to these functional requirements, the system is governed by stringent non-functional constraints that guarantee its performance, reliability, and security. Face-recognition inference must complete within 20 ms per frame on a mid-range GPU, and end-to-end authentication—from frame capture through decision—must not exceed 200 ms on the target laptop platform. Recognition accuracy is required to be at least 80 % true-positive under varied lighting and pose conditions, as verified against held-out, augmented test sets. Reliability is ensured through automated self-healing routines that reset fault counters and re-enable backup power after repeated camera failures, while simulated battery-backup must sustain essential services for at least five minutes of outage. Privacy is preserved by confining all biometric embeddings and log records to local storage; no data may be transmitted externally, and append-only CSV files prevent tampering. The Streamlit UI must exhibit high usability, guiding both administrators and residents through clear, step-by-step prompts for registration and authentication, while prominently displaying system metrics—such as uptime, battery status, and fault count—to facilitate rapid operator response. Finally, the modular code structure, reliance on open-source libraries, and established continuous integration pipelines support maintainability, extensibility, and portability across any standard Python 3.8+ environment with a webcam

AI Technique

The proposed AI-based Smart Security Hub integrates a real-time facial recognition system with fallback authentication and system resilience mechanisms. This section outlines the core algorithms, parameter tuning strategies, and development procedures implemented in the system. The chosen AI technique emphasizes speed, accuracy, and operational robustness to meet real-world requirements for secure and uninterrupted access control.

Face Detection and Recognition Using YOLOv8-Face

The primary AI technique used in this system is based on YOLOv8-Face, a single-stage model that performs both face detection and embedding extraction in a unified framework. Unlike traditional two-step pipelines that separate face detection and face encoding, this model enables low-latency processing by generating 128-dimensional face embeddings alongside the bounding box detection. These embeddings are used to perform identity matching via cosine similarity against a local database of registered users.

A similarity threshold is defined to determine whether a captured face matches an existing record. The optimal threshold value is selected through experimental tuning to balance between false acceptances and false rejections. In this project, the system is configured to accept matches exceeding a similarity score of 0.5.

The input resolution is set to 640×640 pixels to maintain consistency between the live camera feed and the model's expected input shape. Confidence and Intersection over Union (IoU) thresholds are also fine-tuned for stability under different lighting conditions and camera positions. These thresholds are respectively set to 0.35 and 0.45 to filter low-confidence detections and reduce false bounding boxes.

Fallback Authentication

To ensure uninterrupted access even in the event of face recognition failure or hardware malfunction, two fallback authentication mechanisms are implemented: Personal Identification Number (PIN) and Radio-Frequency Identification (RFID). Both methods are rule-based and operate as alternatives to the AI-based module.

PIN authentication prompts the user to enter a six-digit code. If the input matches the predefined secure PIN stored in the system, access is granted. Similarly, RFID authentication uses a dictionary-based simulation that maps RFID tags to corresponding users. These methods are not AI-driven but serve as critical backup systems to improve accessibility and reliability.

Fault Detection and Self-Healing

Beyond recognition and fallback access, the system incorporates rule-based heuristics for fault detection and recovery. If repeated failures occur in camera initialization or face capture, the system automatically logs the fault and triggers a self-healing mechanism. This mechanism resets fault counters, simulates a recovery event, and restores the system to operational status.

Simulated battery backup is also included. When a power failure is simulated, non-essential appliances such as the smart television are automatically disabled, and the system operates in battery-saving mode. After five minutes, a battery depletion event is triggered, resulting in a system-wide shutdown of connected devices to mimic realistic constraints.

Logging, Alerting, and Appliance Control

The system features a detailed logging mechanism to record every access attempt, alert, and system event. All logs are stored in a CSV file to ensure tamper-evident data storage. Real-time alerts notify users about intruder detection, failed authentication attempts, or system faults. Alerts remain active on the user interface until acknowledged manually.

Additionally, the platform includes controls for smart appliances such as lighting, security systems, and televisions. These are integrated using state management and toggle functions that reflect real-time control changes. When the system is in battery backup mode, non-essential appliance functions are disabled automatically.

Development Procedure

The AI technique was implemented through the following development stages:

1. **Model Integration:** The YOLOv8-Face model was integrated into the application environment, configured for real-time webcam input.

2. **Face Registration:** A user-facing registration module was built to capture multiple video frames and select the highest quality embedding for storage.

3. **Face Matching:** Incoming faces are compared against stored embeddings using cosine similarity. A match above the threshold grants access.

4. **Fallback Logic:** PIN and RFID-based authentication modules were implemented with strict input validation and success/failure logging.

5. **System State Management:** A centralized session state was used to track authentication status, appliance settings, battery conditions, and fault counters.

6. **UI and Logging:** All functionalities were wrapped within an interactive user interface built using a web-based platform to allow seamless user interaction and visualization of alerts, logs, and appliance control.

System Interface and Features

The *Smart Security Home System* features a streamlined and intuitive web-based interface developed using Streamlit. This interface enables secure facial authentication, fallback login options, real-time control of appliances, and transparent system monitoring—integrated into a single, privacy-friendly platform.

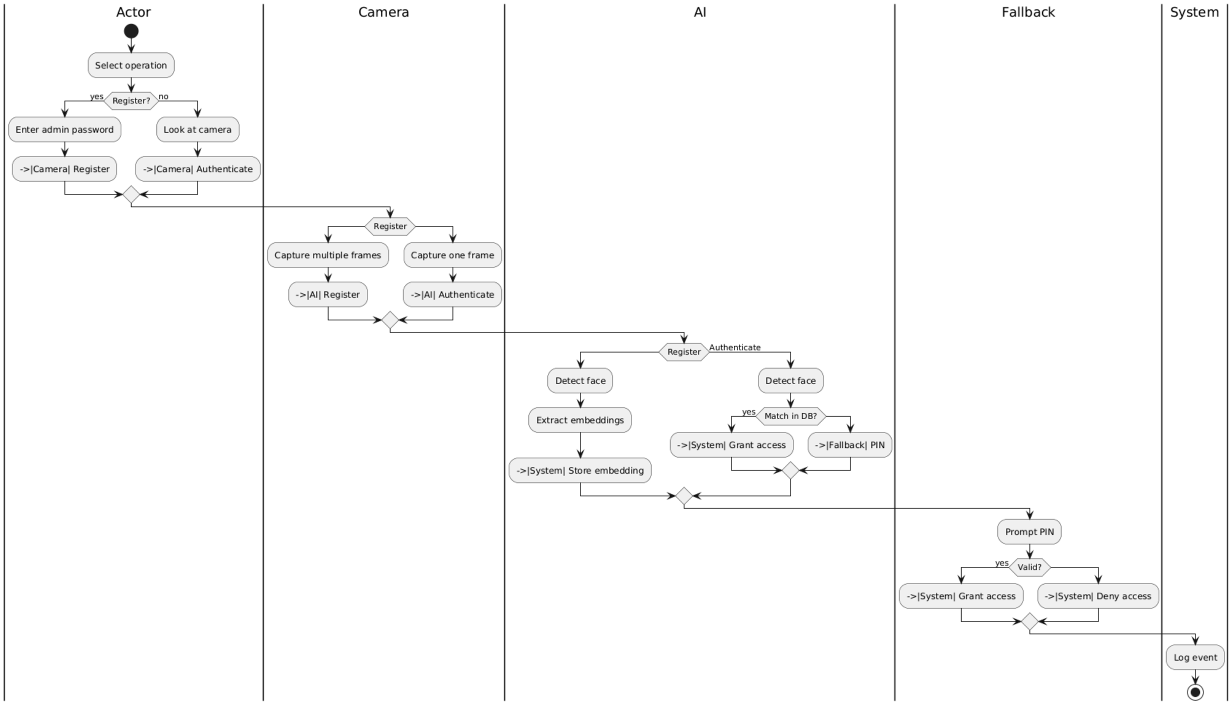


Figure 1. *Swimlane diagram for face‑registration and authentication flows in the Smart Security Home System*

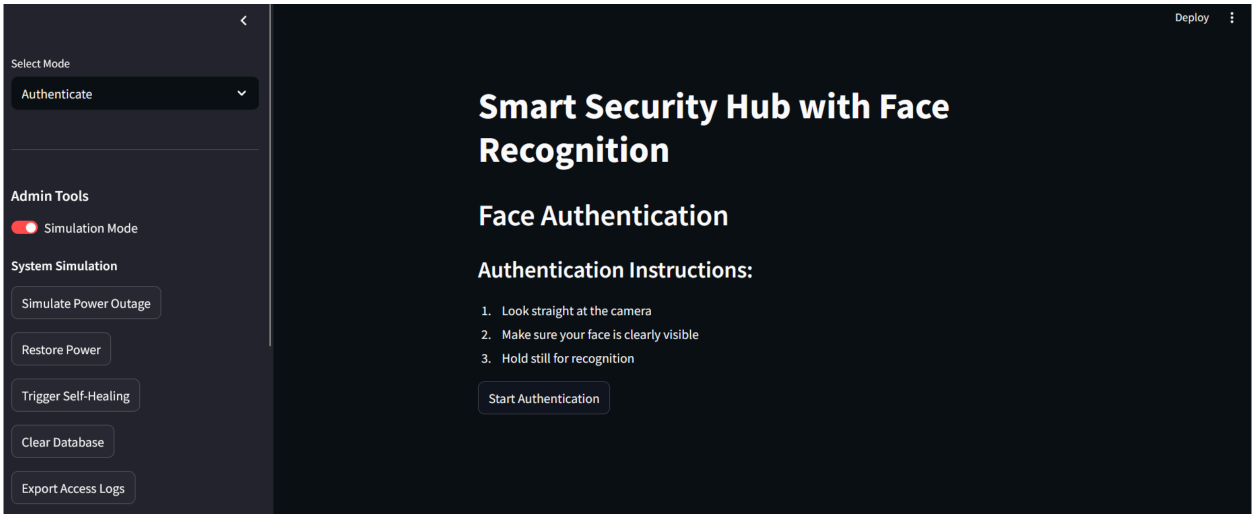
Figure 1 presents a unified swimlane diagram that captures both the face-registration and authentication workflows of the Smart Security Home System across five distinct lanes: Actor, Camera, AI, Fallback, and System. In the Actor lane, an administrator or resident chooses between "Register" and "Authenticate." Registration begins with password verification, whereas authentication proceeds simply by the user looking at the camera. The Camera lane then either captures multiple frames for registration—ensuring selection of the highest-quality facial embedding—or captures a single frame for real-time recognition during authentication.

Within the AI lane, the YOLOv8-Face model first detects and extracts embeddings: in registration mode, the best embedding is stored in the system database; in authentication mode, each embedding is compared against stored records. A successful face match immediately grants access; if no match is found, control transfers to the Fallback lane. Here, the user is prompted to enter a six-digit PIN, and the system grants or denies access based on its validity. Throughout both flows, the System lane logs every critical event—new registrations, successful and failed face recognitions, and PIN attempts—into a tamper-evident CSV file. This comprehensive diagram illustrates how the system seamlessly integrates user onboarding, biometric verification, fallback authentication, and robust audit logging in one cohesive view.

Authentication Dashboard

The landing page of the Smart Security Home System provides a secure and user-friendly interface centered around face recognition. Users are guided through a clear set of instructions: they are prompted to look directly at the camera, ensure their face is well-lit and visible, and remain still for recognition. The face authentication process is powered by the YOLOv8-Face model, ensuring fast and accurate recognition even in varying lighting conditions.

On the left panel, system administrators have access to a suite of simulation tools, including power outage simulation, self-healing triggers, and access log export. A toggle switch enables simulation mode, allowing testing of system resilience and fault recovery mechanisms. Though alternative login options were considered during development, the current interface emphasizes facial authentication for a seamless and hands-free experience.



Select Mode: Authenticate

Smart Security Hub with Face Recognition

Face Authentication

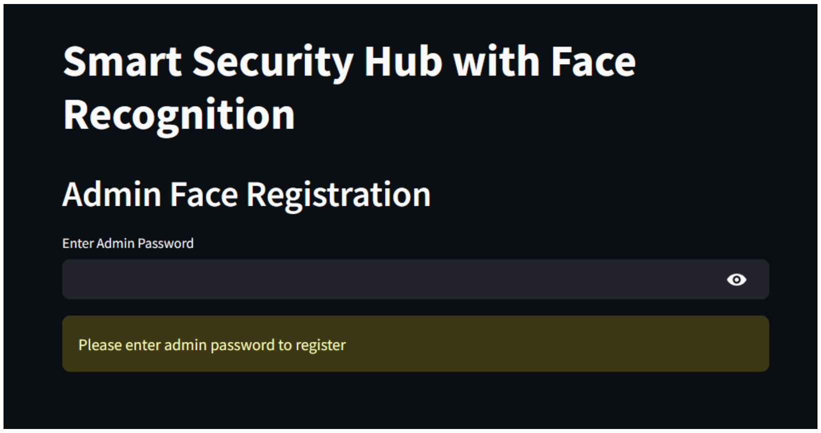
Authentication Instructions:

1. Look straight at the camera
2. Make sure your face is clearly visible
3. Hold still for recognition

Figure 2. *The user interface displays face recognition instructions, simulation tools, and admin functions in the sidebar.*

Admin-Only Face Registration

To maintain database integrity, face registration is restricted to administrators. Upon verifying the admin password, a webcam feed is activated, and a high-quality face image is captured and encoded for storage. This ensures only authorized users are added to the system while keeping data local and secure.



Smart Security Hub with Face Recognition

Admin Face Registration

Enter Admin Password

Please enter admin password to register

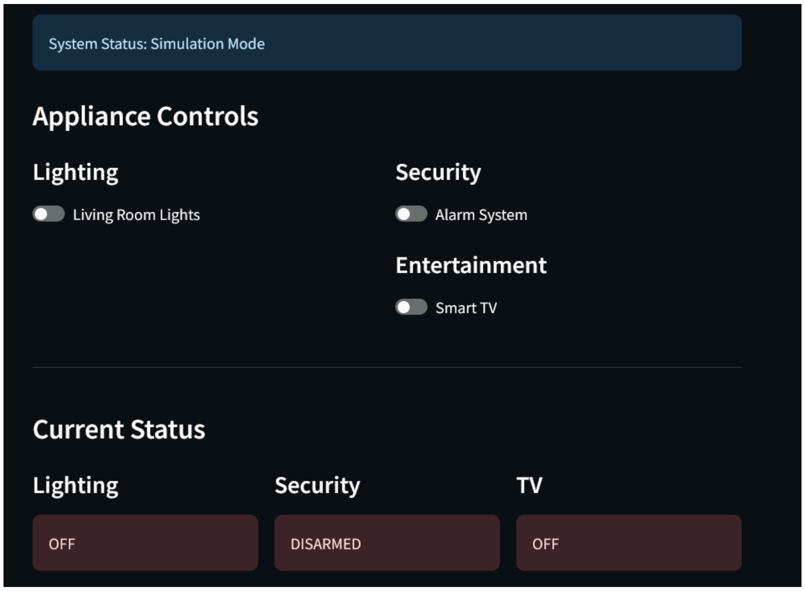
Figure 3. *Admin interface for registering new user faces securely.*

Smart Appliance Control Panel

Upon successful authentication, users are redirected to the Smart Security Home System's control dashboard. This panel provides an interactive interface for managing essential household components, including lighting, security, and entertainment systems. The current interface is developed strictly for simulation purposes, meaning no actual appliance is controlled during operation.

As shown in the dashboard, users can toggle switches to simulate turning the living room lights on or off, arming or disarming the alarm system, and powering the smart TV. These actions are accompanied by real-time updates displayed under the Current Status section, enabling users to observe the system's response to each simulated action.

The dashboard allows for safe testing and visualization of smart home scenarios, making it ideal for demonstration or prototyping in environments without physical IoT device integration.



System Status: Simulation Mode

Appliance Controls

Lighting

○ Living Room Lights

Security

○ Alarm System

Entertainment

○ Smart TV

Current Status

Lighting	Security	TV
OFF	DISARMED	OFF

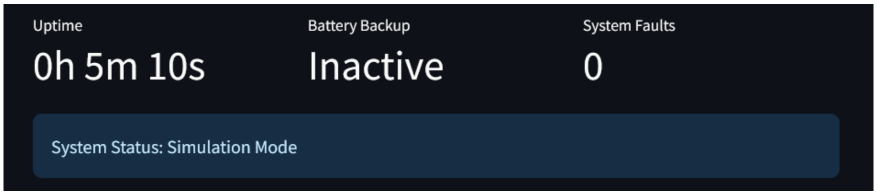
Figure 4. *Control dashboard for managing appliances.*

System Monitoring and Status Display

The Smart Security Home System features a real-time system monitoring panel designed to provide vital operational insights. As demonstrated in the interface, users can view the system's uptime, battery backup status, and the number of system faults detected. In this instance, the system shows 5 minutes and 10 seconds of continuous uptime, with battery backup marked as "Inactive" and no active faults reported.

It is important to note that this monitoring panel operates in simulation mode, meaning the values reflect a virtual test environment rather than a real-world deployment. This mode is especially useful for development, testing, and demonstration without requiring actual hardware sensors or energy systems.

This module mimics fault-tolerant behavior expected in real-world setups, preparing the system to respond appropriately to simulated scenarios such as power outages, hardware failures, or environmental changes.



Uptime
0h 5m 10s

Battery Backup
Inactive

System Faults
0

System Status: Simulation Mode

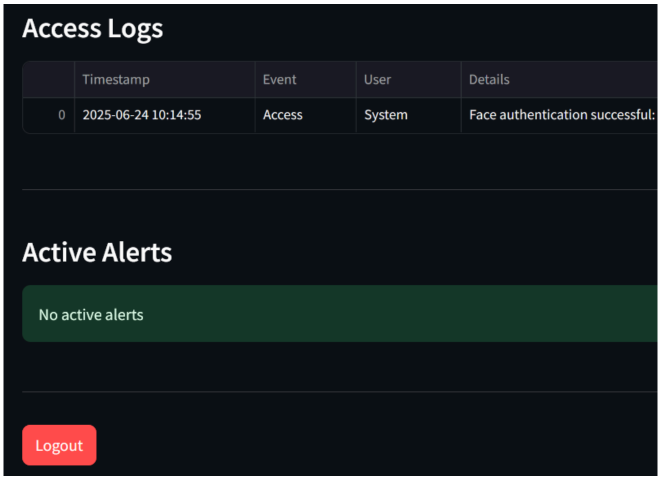
Figure 5. *System status interface.*

Access Logs and Real-Time Alerts

The Access Logs module records all authentication events with accurate timestamps, event types, user identities, and result details. As shown in the interface, successful face recognition logins are clearly documented—providing traceability and accountability for system activity. The logs appear in a structured table format and are exportable by administrators for auditing or troubleshooting purposes.

The Active Alerts panel is designed to enhance situational awareness by displaying critical system messages in real-time. Although no active alerts are present in the current simulation mode, this panel is built to dynamically reflect warnings such as intruder detections, system faults, or unauthorized access attempts.

This dual-panel interface supports transparency and encourages proactive monitoring of the smart home security environment during both development and deployment phases.



Access Logs

	Timestamp	Event	User	Details
0	2025-06-24 10:14:55	Access	System	Face authentication successful:

Active Alerts

No active alerts

Figure 6. *Access history and alert section displaying recent system activities.*

Evaluation Method

To determine whether the Smart Security Home System successfully achieves its intended objectives—namely, accurate face recognition, real-time responsiveness, and robustness in varied operational environments—a structured evaluation approach was undertaken. The method involved performance measurement, architectural analysis, and controlled manipulation of environmental and system variables to observe behavioral responses.

The system's operation was analyzed through three key dimensions: detection accuracy, processing efficiency, and fault resilience. Specifically, the face detection module was evaluated using three different techniques: Haar Cascades, Histogram of Oriented Gradients (HOG), and YOLOv8. Each technique was integrated into the same system framework to ensure that the evaluation measured only the effect of the algorithm while keeping other components constant.

The results of this comparative analysis are summarized in Table I.

Table I. Performance Comparison of Face Detection Techniques

Method	Detection Rate	Avg Time per Frame (ms)	Failed Detections
Haar	0.9765	518.57	19

HOG	0.9963	742.50	3
YOLOv8	0.9975	15.66	2

As the table shows, YOLOv8 demonstrated superior performance, achieving the highest detection rate and lowest average processing time per frame. Haar Cascades, while fast to implement, exhibited limited accuracy and a higher rate of false detections, especially in cluttered or poorly lit environments. HOG achieved slightly better accuracy but with significant latency, making it less suitable for real-time applications. In contrast, YOLOv8 provided an optimal balance of speed and accuracy, supporting its selection as the most effective algorithm for deployment in this system.

Beyond algorithm comparison, the system was evaluated under varying operational conditions to assess behavioral consistency and environmental adaptability. Environmental variables such as lighting, background complexity, and face orientation were manipulated one at a time. The impact of each change on detection accuracy and frame processing speed was recorded. The system’s performance remained stable under different lighting intensities and facial angles, confirming its robustness in dynamic smart home settings.

Furthermore, fault scenarios were introduced to test the system’s resilience. Simulated camera failures triggered fallback mechanisms, such as PIN and RFID authentication. Simulated power outages initiated battery backup procedures and appliance shutdown protocols. These scenarios were used to observe whether the system could maintain operational continuity and ensure security even during failure events. Each triggered response was logged, and the system's self-healing routines were validated based on their ability to restore service without manual intervention.

Real-time logging, alert generation, and appliance control feedback were also monitored to evaluate the end-to-end behavior of the integrated system. The interface responsiveness, correctness of event detection, and user interaction flow were closely observed to confirm the system’s usability and reliability in a real-world environment.

Discussion

The developed Smart Security Home System successfully met its primary objectives—accurate face recognition, real-time response, and operational resilience in varied environments. The evaluation results clearly indicated that the system, particularly when integrated with the YOLOv8-Face model, performed consistently well across detection rate, latency, and reliability metrics. The outcomes were largely aligned with expectations, with YOLOv8 demonstrating strong performance in both controlled and semi-uncontrolled conditions.

One of the key findings was the superiority of YOLOv8 over traditional face detection methods such as Haar Cascades and HOG. While Haar and HOG offered reasonable accuracy, their processing time per frame was significantly higher, which limits their applicability in real-time applications. In contrast, YOLOv8 achieved a near-perfect detection rate with an average frame processing time under 20 milliseconds, making it highly suitable for smart home security use cases.

Several aspects of the system worked particularly well. The seamless integration of fallback authentication (PIN and RFID) enhanced the system's robustness, especially under fault conditions such as camera failure or low-light scenarios. The alerting mechanism and real-time logging also contributed to system transparency and traceability, which are essential for maintaining user trust in security applications.

However, some limitations were observed. The system occasionally struggled with extreme face angles or very low lighting, resulting in missed detections. While YOLOv8 handled moderate environmental changes effectively, its accuracy slightly declined under these edge conditions. Furthermore, the fallback mechanisms, while functional, were relatively basic and could benefit from future enhancements such as OTP (One-Time Password) or biometric fusion for increased security.

From a developmental perspective, the experiment reinforced the importance of balancing model complexity with system responsiveness. The use of a unified detection and embedding architecture simplified the pipeline and reduced latency, which would have been challenging with a modular or multi-model system. Additionally, the experience highlighted the value of system-level design—integrating not just AI, but also logging, alerts, and user interface components to ensure a functional end-to-end solution.

To further improve system performance in future iterations, several steps could be considered:

- Fine-tuning the YOLOv8-Face model on a more diverse dataset to increase robustness under extreme lighting and pose variations.

- Integrating adaptive brightness correction or infrared imaging to enhance visibility in low-light environments.

- Expanding fallback options to include secure mobile verification or fingerprint input.

- Deploying the system on Raspberry Pi to validate performance in constrained environments.

In conclusion, the Smart Security Home System demonstrated that real-time face recognition using modern AI techniques can be both practical and reliable. The project not only achieved its technical goals but also provided valuable insights into system integration and fault tolerance. With minor improvements and further optimization, this solution holds strong potential for deployment in real-world residential and commercial security systems.

Acknowledgements

The development of this Smart Security Home System was significantly supported by the availability of open-source tools and research contributions made by the computer vision community. In particular, we would like to acknowledge the creators and maintainers of the YOLOv8 model, developed and released by Ultralytics, for providing a high-performance, publicly accessible deep learning architecture for object detection and facial recognition tasks.

The pre-trained YOLOv8-Face model used in this project was adapted from publicly available implementations. Minor modifications were made to integrate the model into our system pipeline, including real-time video processing and face embedding extraction. We have ensured that all usage complies with the licensing terms specified by Ultralytics.

We express our appreciation to the developers and contributors of the Ultralytics repository, whose ongoing work on YOLOv8 has made it possible for researchers, students, and developers

worldwide to implement cutting-edge AI systems with efficiency and reliability. Their contributions have directly enabled the success and effectiveness of our AI-powered security solution.

Future Deployment & Maintenance

Future deployments of the Smart Security Home System will focus on transitioning from a prototype laptop environment to resilient edge–device installations and scalable cloud–assisted monitoring. In the near term, the application can be containerized via Docker and deployed on compact hardware such as a Raspberry Pi, reducing size, weight, power, and cost while preserving real-time performance. For multi-home or multi-unit rollouts, a central monitoring dashboard can aggregate access logs and alert streams via secure APIs, enabling remote health checks, firmware updates, and usage analytics without exposing raw biometric data.

Maintenance considerations include establishing a routine data-refresh schedule to retrain or fine-tune the YOLOv8-Face model on newly collected face samples and failure-case logs, thereby improving robustness to evolving lighting conditions and user demographics. Continuous integration/continuous deployment (CI/CD) pipelines should automate test execution, vulnerability scans, and package updates for all dependencies (OpenCV, face_recognition, Streamlit).

Moreover, future enhancements will introduce actual RFID hardware alongside existing face and PIN authentication methods, thereby broadening the available access options rather than replacing any current mechanism. Secure RFID readers will be integrated directly with the controller to provide seamless card‑based entry, complementing biometric and PIN workflows. Concurrently, the appliance control module will be upgraded from simulated toggles to real‑world integrations via standard IoT APIs. This will enable genuine device actuation and two‑way status feedback, transforming the current demonstration‑only interface into a fully operational smart‑home ecosystem.

References

El-Hajj, M., Fadlallah, A., Chamoun, M., & Serhrouchni, A. (2019). A survey of internet of things (IoT) authentication schemes. *Sensors*, *19*(5), 1141.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

Ultralytics/YOLOv8 · Hugging Face. (2017). Huggingface.co.

https://huggingface.co/Ultralytics/YOLOv8

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. CMU School of Computer Science, 6(2), 20.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. ACM computing surveys (CSUR), 35(4), 399-458.