Sensors and Materials, Vol. 38, No. 8 (2026) 4549–4562
MYU Tokyo

S & M 4581

An Edge-based Intelligent Sensing System for Access Control with Face Recognition and Temporal Suspicious-behavior Detection

Nai-Ci Chen, Ruo-Ya Chao, and Wei-Fu Lu*

Department of Communications Engineering, Feng Chia University,
No. 100, Wenhua Rd., Xitun Dist., Taichung City 407102, Taiwan (R.O.C.)

(Received November 18, 2025; accepted July 27, 2026)

Keywords: smart access control, face recognition, suspicious-behavior detection, dwell-time analysis, embedded systems

In this study, we present an edge-based intelligent sensing system for access control that integrates real-time face recognition with an embedded suspicious-behavior detection module. Access control serves as the application scenario for this edge-based sensing framework, in which camera-captured observations are processed locally into identity- and behavior-level security decisions. The system performs dynamic user registration and classifies individuals as owners, guests, or strangers, each triggering a dedicated response. Two complementary strategies identify suspicious strangers: dwell-time accumulation near entrances and repeated-appearance monitoring, which flags an individual reappearing more than ten times within 10 min, capturing loiterers who evade dwell-time-only thresholds. The hardware platform is a low-cost Raspberry Pi, with a Python/OpenCV (Open Source Computer Vision Lib45rary) pipeline employing the Local Binary Pattern Histogram algorithm for face recognition. Experimental evaluation demonstrates recognition accuracies of 100, 95.56, and 95.23% for owners, guests, and strangers, respectively, with an average recognition latency of 0.39 s and a processing throughput of 12.87 frames per second. The suspicious-behavior detection module attains 95.10% accuracy with zero false alarms, confirming the feasibility of the proposed solution for edge-based intelligent security applications. The evaluation was conducted under controlled, single-entry laboratory conditions; future work will extend the system to multi-camera deployments and evaluate robustness under varying facial expressions, poses, and occlusions.

1. Introduction

With the growing global demand for robust security and operational efficiency in modern facilities, AI-driven surveillance and access control systems have become indispensable. These intelligent solutions offer significant advantages over traditional methods such as keycards or manual inspection, which are often susceptible to human error, operational delays, and physical security risks. While a new generation of solutions now supports automated identity verification

*Corresponding author: e-mail: wflu@o365.fcu.edu.tw
https://doi.org/10.18494/SAM6037

ISSN 0914-4935

and proactive anomaly detection, their widespread adoption is often hindered by several key limitations. Specifically, many commercial access and monitoring systems rely heavily on powerful cloud services or high-performance servers for intensive image and video analysis. This dependency leads to high hardware and maintenance costs, introduces potential network latency that compromises real-time responsiveness, and raises significant privacy concerns by transmitting sensitive data to external servers.

A critical gap also exists in the functionality of many current systems. Even those that incorporate advanced face recognition often make only static judgments based on single-frame images, lacking the ability to build and analyze long-term behavioral context. This deficiency makes them vulnerable to a simple but effective form of evasion, where suspicious individuals briefly leave and then re-enter the monitored area to reset a static detection timer. Another common blind spot arises when strangers repeatedly appear in front of an entrance but only for short intervals, avoiding conventional dwell-time thresholds while still exhibiting abnormal behavior. These scenarios represent significant security weaknesses that most existing systems fail to address effectively.

To overcome these limitations, we propose a novel, low-cost smart access control system designed to operate efficiently on an embedded platform, integrating a vision sensor with embedded, real-time behavioral analysis to advance intelligent, sensor-driven access control. Our research is motivated by the need for a security solution that is not only cost-effective and real-time, but also intelligent enough to analyze a subject's behavior over time, even across noncontiguous events. We aim to demonstrate the feasibility and deployment flexibility of building a sophisticated security system within a resource-constrained environment by performing all core computational tasks at the edge.

It is worth emphasizing that access control is treated in this work as an application domain rather than as the object of study itself. The essential contribution is a general camera-based edge-sensing pipeline—capturing visual data, identifying individuals, and accumulating their behavior over time to produce identity-level and temporal behavior-level decisions—that is instantiated here for door access. Access control is adopted primarily because it offers a well-defined three-way decision structure (owner/guest/stranger) with a clear ground truth for evaluating recognition and behavioral-detection accuracy; in principle, the same pipeline can be redirected toward other identity- and behavior-aware monitoring tasks.

The main contributions of this paper are summarized as follows:

- A Low-Cost, Integrated System Prototype: We successfully designed and implemented a complete smart access control system on a single Raspberry Pi platform, leveraging Python and OpenCV (Open Source Computer Vision Library). This approach demonstrates a practical and affordable alternative to high-end commercial solutions that rely on expensive hardware or cloud services.
- Dual Suspicious-behavior Detection Mechanisms: In addition to a dwell-time accumulation method that tracks how long an unidentified person remains within the camera's view, even across temporary exits and re-entries, we introduce a repeated-appearance rule that flags a stranger as suspicious if they are detected more than ten times within a 10 min window. Together, these mechanisms address both prolonged loitering and frequent short-term reappearances, closing critical evasion loopholes in existing systems.

- High Performance on an Embedded Platform: Our preliminary experiments confirmed that the system achieves high accuracy and low latency for both face recognition and suspicious detection, with an impressive zero false alarm rate for suspicious activity. This proves that sophisticated security functions can be reliably executed on resource-limited hardware.
- Comprehensive Security Framework: The proposed system incorporates multi-factor authentication via a mobile app and multi-layered security measures, including data encryption and alert mechanisms, providing a holistic and robust security solution.

The remainder of this paper is organized as follows. In Sect. 2, we review related work on face recognition, behavioral analysis, and edge-based access control. In Sect. 3, we describe the overall system architecture and the methodology of each functional module, including facial recognition, suspicious-behavior detection, and security mechanisms. In Sect. 4, we present the experimental setup and performance evaluation. In Sect. 5, we discuss the experimental results and their implications. Finally, in Sect. 6, we conclude the paper and outline possible directions for future work.

2. Related Work

2.1 Face recognition and behavioral anomaly detection

Face recognition has long been a cornerstone of intelligent access control. Classical, lightweight approaches such as the Local Binary Pattern Histogram (LBPH) algorithm remain attractive for embedded devices because of their computational efficiency and robustness to monotonic illumination changes.[1] With the rise of deep learning, methods such as DeepFace, FaceNet, and ArcFace introduced discriminative embeddings and margin-based learning objectives that markedly improved recognition under pose and lighting variations.[2–4] In practice, modern pipelines often combine robust face detection and alignment networks—for example, Multi-Task Cascaded Convolutional Networks (MTCNNs), which jointly localize face regions and key landmarks,[5] with deep embedding models, ensuring that recognition is stable even in unconstrained, real-world scenes. Comprehensive surveys further document the transition from handcrafted features to end-to-end deep representations, highlighting the accuracy–compute trade-offs that continue to motivate efficient or hybrid solutions for deployment at the edge.[6]

In parallel, access control surveillance has increasingly emphasized behavioral analysis to detect abnormal activities before security incidents occur. A prominent line of work targets potential abnormal behaviors via trajectory cues; for example, Núñez *et al.* detected loiterers by analyzing directionality, pace, and time-based persistence on annotated sequences.[7]

At the crowd level, formulations such as the Social Force Model[8] and its computer-vision adaptation for abnormal crowd behavior[9] illustrate how abnormality emerges not from a single frame but from temporal interaction patterns. Group-level pedestrian behavior modeling, such as inferring walking patterns from interactions with stationary crowd groups, further illustrates how crowd context can inform behavior prediction and anomaly detection.[10]

More broadly, video anomaly detection has evolved from pipelines based on handcrafted motion or appearance descriptors to deep spatiotemporal learning. Relevant studies note

remaining challenges in data sparsity, domain shift, and evaluation methodology.[11–13] Complementary approaches such as skeleton-based pose estimation (e.g., OpenPose) provide additional cues for recognizing human activity under occlusion or in crowded settings.[14]

These studies collectively highlight a key insight: abnormality is often revealed through temporal patterns rather than instantaneous observations. However, most prior work has focused either on large-scale crowds or on single-session dwell-time analysis, with comparatively little attention given to the repeated short-term reappearances of the same individual.

This represents a blind spot in the context of access control, as a stranger may evade a dwell-time threshold by leaving and re-entering multiple times within a short period. To address this limitation, our system incorporates the ability to dynamically learn and recognize if a stranger who leaves the frame and reappears is the same person. This allows us to accumulate the stranger's dwell time and count their repeated appearances within a fixed time frame. By doing so, we extend anomaly detection concepts from the crowd domain to the more granular, individual-focused environment of smart access control.

Finally, driven by the need for cost-effectiveness, privacy protection, and faster response times, the deployment of access control systems has shifted from cloud-based to edge-based processing. The Raspberry Pi ecosystem offers a low-cost, I/O-rich platform ideal for integrating cameras, relays, and alarms in real-world deployments.[15] In our prototype, this ecosystem is realized through a Raspberry Pi Camera Module 3 NoIR as the vision sensor, a General-Purpose Input/Output (GPIO)-triggered relay module driving the electronic door-lock solenoid, and a GPIO-triggered buzzer/siren as the alarm actuator. Possible improvements include adding infrared illumination or a dedicated low-light sensor for more reliable night-time recognition, and using a latching relay with fail-safe behavior to preserve door security during power loss. It remains challenging to achieve both high-accuracy recognition and robust suspicious-behavior detection on such resource-constrained hardware. However, this is precisely what motivated our design: an embedded access control system that combines efficient recognition with temporal behavior analysis to provide actionable, real-time security decisions at the edge.

2.2 Comparison with related systems

To clarify the novelty of the proposed system relative to prior work, Table 1 shows representative studies along four dimensions: identity-level recognition, temporal behavioral analysis, whether such analysis persists across a person's noncontiguous re-entries, and integration with an access-control response.

As summarized in Table 1, existing face-recognition studies[1–6] emphasize identity verification without behavioral analysis, while trajectory-based loitering detection[7] and crowd- or scene-level anomaly detection[8–13] analyze temporal or crowd behavior without identity recognition, persistent per-individual tracking, or access-control integration. In contrast, the proposed system treats access control as an application scenario of edge-based visual sensing: its novelty does not lie in the use of face recognition alone, but in the integration of identity-level recognition and temporal behavior-level sensing—via dwell-time accumulation and repeated-appearance detection that persist across a stranger's re-entries—on a low-cost embedded

Table 1
Comparison of the proposed system with representative related work

Study/approach	Identity recog.	Temporal behavior	Persists re-entry	Access control	Platform
Face recognition methods[1–6]	Yes	No	—	No	Server/GPU
Loitering detection via trajectory analysis[7]	No	Yes (single session)	No	No	Surv. server
Crowd-level flow/force models[8,9]	No	Yes (crowd-level)	No	No	Surv. server
Stationary-crowd behavior modeling[10]	No	Yes (crowd-level)	No	No	Server
Deep scene-level anomaly detection[11–13]	No	Yes (scene-level)	No	No	GPU server
Proposed system	Yes	Yes (per-individual)	Yes	Yes	Raspberry Pi (edge)

platform. This enables the system to respond not only to recognized owners and guests, but also to suspicious patterns exhibited by strangers near an entrance. A direct quantitative comparison with other recently reported systems is difficult because published works differ substantially in hardware, dataset, and evaluation protocol; nonetheless, the recognition latency (0.39 s) and throughput (12.87 FPS) achieved here on a Raspberry Pi 5 are competitive with LBPH-based systems reported in the literature, while the zero-false-alarm rate for suspicious-behavior detection compares favorably with trajectory-based loitering detectors evaluated on unconstrained video.

3. System Architecture and Methodology

This smart access control system is designed with a modular architecture to ensure real-time performance, scalability, and reliability on a resource-constrained embedded platform. The architecture integrates four major layers—sensing, recognition, behavioral analysis, and response—into a cohesive access solution suitable for embedded environments.

3.1 System architecture

As illustrated in Fig. 1, the system is implemented on a Raspberry Pi platform, which functions as the central processing core. It integrates multiple modules, including image acquisition, face recognition, suspicious-behavior detection, user notification, and device control. These modules operate in concert to provide layered access management, real-time decision-making, and responsive security alerts.

The image acquisition module captures continuous video streams through the Raspberry Pi Camera Module. Incoming frames are processed by the image recognition module, implemented in Python with OpenCV. The system employs the LBPH algorithm for robust face recognition under constrained hardware.

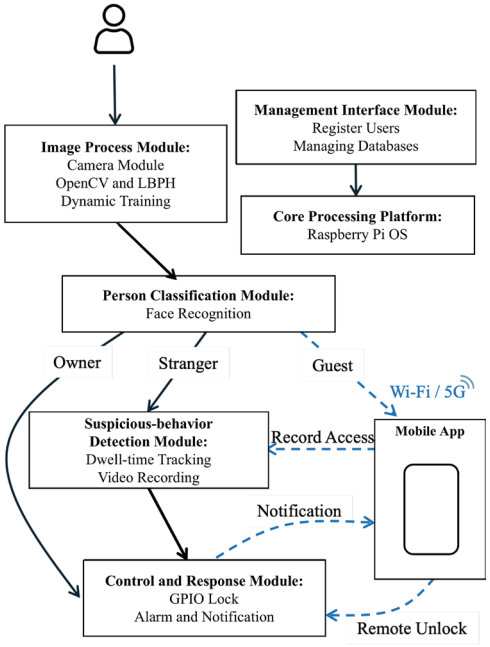

Fig. 1. (Color online) Architecture of the proposed smart access control system. Solid arrows denote data processing flows; dashed arrows indicate Wi-Fi/5G communication with the mobile app.

Dynamic enrollment allows new users to be registered in real time, building feature profiles directly from captured video streams. Dynamic training is supported, enabling new users to be enrolled directly from captured video streams. This capability allows the system to adaptively expand its database without requiring offline preprocessing. Once recognition is performed, the person classification module assigns the individual to one of three categories:

1. Owner – Registered users, granted immediate access via the automated unlocking of the door lock.
2. Guest – Pre-authorized visitors whose entry requires real-time approval through the owner's mobile application.
3. Stranger – Unregistered individuals who are denied access and evaluated by the suspicious-behavior detection module.

On the control side, the response module interfaces with the Raspberry Pi GPIO to operate the electronic door lock and activate alarms when unauthorized access is attempted. The mobile application, connected via Wi-Fi/5G, acts as the remote user interface, providing live video feeds, recognition outcomes, event notifications, and access control. A management interface further allows administrators to register new users, manage facial databases, and configure security policies.

3.2 Suspicious-behavior detection

To overcome the limitations of conventional single-frame analysis, which lacks the ability to capture temporal behavioral patterns, the system incorporates two complementary mechanisms

for detecting suspicious individuals. Both approaches rely on the dynamic enrollment of unregistered persons, which enables the system to consistently recognize the same individual across multiple entries and exits.

(a) Dwell-time Accumulation

For each unregistered person, the system maintains a running dwell-time counter that increases whenever the person is visible and pauses when they leave the camera's field. The total accumulated dwell time after N observed appearances is given by

$$T_N = \sum_{k=1}^{N} \left(t_{exit}^{k} - t_{enter}^{k} \right), \tag{1}$$

where t_{enter}^{k} and t_{exit}^{k} denote the entry and exit timestamps of the k-th appearance, respectively. An alert is triggered once T_N exceeds the dwell-time threshold θ_{dwell} (e.g., 60 s). In other words, if a stranger stays in front of the camera for too long in total, even across multiple short appearances, the system identifies the individual as suspicious.

(b) Repeated-appearance Detection

In addition, for each unregistered person, the system maintains a running counter that records how many times the same individual has appeared in front of the camera. When person i is detected for the first time, the counter is initialized to 1. Each time the person leaves and re-enters the camera's field, the counter is incremented by one. Let t_{enter}^{k} be the entry time of the k-th appearance of an unregistered person. An alert is triggered if the condition in Eq. (2) holds:

$$t_{enter}^{k} - t_{enter}^{k-\theta_{count}+1} \le \Delta T, \quad for\ each\ k \ge \theta_{count}. \tag{2}$$

This means that the most recent θ_{count} appearances (e.g., 10 appearances) have all occurred within the last ΔT time units (e.g., 10 min). In other words, if a stranger reappears too frequently within a short observation window, exceeding the count threshold θ_{count}, the system flags this behavior as suspicious, typically corresponding to loitering or repeatedly scanning the entrance with potential malicious intent.

3.3 Information security and safety mechanisms

To guarantee data integrity and safeguard user privacy, the system incorporates a multi-layered security framework. All facial recognition data and access logs are stored in encrypted form, preventing unauthorized retrieval or manipulation. This ensures that biometric data remain protected even under constrained hardware deployment.

In addition to encryption, the system enforces multi-factor authentication. Facial recognition alone is insufficient for granting access; guest entries must be explicitly authorized through the owner's mobile application. Each approval is time-stamped and securely logged, enabling traceability and accountability in all access events.

Communication between the Raspberry Pi and mobile devices is carried out over secure Wi-Fi/5G channels. To prevent unauthorized interception, device binding and authentication protocols are applied so that only registered mobile clients are permitted to receive alerts and access logs. This design provides resilience against both external network threats and local tampering attempts.

3.4 Emergency response mechanism

In scenarios involving unauthorized access or suspicious behavior, the system provides an immediate escalation pathway to ensure timely intervention. The camera continuously captures and analyzes frames in real time to support recognition and behavioral tracking, and frames are held in a short-term rolling buffer (covering, e.g., the preceding 5 min) that is otherwise continuously overwritten. Once abnormal dwell-time accumulation or repeated-appearance activity is detected, the system commits this rolling buffer to persistent storage as a video clip—capturing the behavior leading up to and including the triggering event rather than only the moment the alert was raised, analogous to pre-event buffering techniques such as dashcam loop recording or smartphone Live Photos—and generates a corresponding structured event log for subsequent review. At the same time, alerts are dispatched to authorized users via the mobile application, ensuring real-time situational awareness.

On the device side, the Raspberry Pi can activate visual or auditory alarms—such as warning lights or sirens—to deter further attempts. In parallel, the system is capable of interfacing with external security infrastructures, enabling integration with broader facility management systems.

Through the mobile application, owners retain remote authority to respond instantly. Emergency actions include the forced locking or unlocking of the electronic door, temporary overrides of access rights, or escalation to security personnel. This layered response not only provides resilience against unauthorized entry but also demonstrates the adaptability of the system for real-world deployment in diverse facility environments.

4. Experimental Setup and Performance Evaluation

After presenting the system design and methodology, in this section, we evaluate the proposed smart access control system through a series of experiments. The evaluation aims to verify the accuracy, latency, and efficiency of the face recognition module as well as the effectiveness of the suspicious-behavior detection mechanisms. In Sect. 4.1, we describe the experimental setup, and in Sect. 4.2, we report the performance evaluation results.

4.1 Experimental setup

The prototype system was evaluated on a Raspberry Pi 5 platform equipped with a Camera Module 3 NoIR. The software environment consisted of Raspberry Pi OS, Python 3.x, and OpenCV 4.x. The experiments included the following:

- testing the recognition accuracy of two owners;
- testing the recognition accuracy of three guests;
- testing seven strangers to evaluate the correct rejection rate and the performance of suspicious-behavior detection.

Each subject underwent 15 independent recognition attempts.

For the enrollment procedure, owners and guests were required to perform a more comprehensive registration process. The system captured a larger set of images to build precise facial feature profiles, ensuring high recognition accuracy in subsequent use. This design choice is motivated by the fact that owner and guest profiles must remain valid across multiple days; once created, their feature data are not limited to same-day recognition but must retain reliability in later sessions. Therefore, more representative images were collected during enrollment to construct robust feature sets.

Table 2 shows the capture frame rate and training time during enrollment. Each owner and guest provided exactly 50 images. "Average FPS" refers to the mean frame rate, in frames per second (FPS), measured while capturing these 50 images, and "Training Time (s)" indicates the time required to train the LBPH model using them.

By contrast, strangers were subject only to a dynamic enrollment procedure. When an unregistered individual first appeared in front of the camera, the system immediately captured images and created a temporary facial profile, which was used for subsequent recognition during the same session. This enrollment lasted only for a short duration and produced coarse feature representations. These temporary features served solely to determine whether repeated appearances belonged to the same individual and were not preserved for long-term recognition. Since their role was limited to supporting suspicious-behavior detection (e.g., dwell-time accumulation and repeated-appearance monitoring), the requirement for feature precision was significantly lower. Consequently, the enrollment strategy for owners/guests and strangers differed substantially.

4.2 Performance evaluation results

System performance was evaluated along two primary dimensions: processing speed and recognition accuracy.

Table 2
Capture frame rate and training time during enrollment.

Category	Average FPS (for 50 images)	Training time (s)
Owner 1	12.85	5.8
Owner 2	12.73	4.2
Guest 1	12.91	4.6
Guest 2	13.04	4.7
Guest 3	12.77	5.5
Average	12.90	4.96

For the face recognition module, both speed and accuracy are critical to access control functionality. Processing efficiency was measured in terms of average time per frame (ms/frame) and FPS on the Raspberry Pi platform. Recognition accuracy was computed as the ratio of correct identifications to total attempts, reported separately for owners, guests, and strangers.

For the suspicious-behavior detection module, experiments were conducted under simulated scenarios such as prolonged dwell times and repeated re-entries within a fixed time window. Two performance indicators were considered:

- Detection Accuracy (Accuracy):

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100\%, \tag{3}$$

- False Alarm Rate (*FAR*):

$$FAR = \frac{FP}{FP + TN} \times 100\%. \tag{4}$$

Here, *TP*, *FP*, *TN*, and *FN* denote true positives, false positives, true negatives, and false negatives, respectively.

Table 3 shows the face recognition performance. The system achieved an average processing throughput of 12.87 FPS on the Raspberry Pi platform, corresponding to an average recognition latency of approximately 0.39 s per image. Recognition accuracy reached 100% for owners, 95.56% for guests, and 95.23% for strangers, confirming the feasibility of real-time access verification in embedded environments.

Table 4 presents the performance of the suspicious-behavior detection module. The detection accuracy was 95.10%, with a false alarm rate of 0%. These results demonstrate that the combined dwell-time accumulation and repeated-appearance monitoring strategies are effective in identifying loitering or repeated short-term reappearances.

Table 3
Face recognition performance.

Metric	Result
Average processing speed	12.87 FPS
Average recognition latency	0.39 s
Accuracy (owners, 2 × 15 tests)	100%
Accuracy (guests, 3 × 15 tests)	95.56%
Accuracy (strangers, 5 × 15 tests)	95.23%

Table 4
Performance of the suspicious-behavior detection module.

Metric	Result (%)
Detection accuracy	95.10
False alarm rate (*FAR*)	0

Note that the triggering time of suspicious-behavior detection inherently depends on cumulative thresholds (e.g., dwell time exceeding 60 s or 10 appearances within 10 min). As a result, the actual alarm may be issued within 1 s after the threshold is crossed, depending on the system's checking interval. This delay does not compromise security, since issuing a notification within a few seconds is sufficient for practical scenarios. Moreover, the magnitude of such delay is strongly affected by the threshold setting. For instance, if the system triggers an alert at 70.1 s, a threshold of 60 s yields a nominal delay of 10.1 s, whereas a threshold of 70 s yields a delay of only 0.1 s. Since the final notification time is identical (70.1 s), the delay value is not a reliable performance indicator. For this reason, triggering latency was not included as a primary evaluation metric in this study.

5. Discussion

The experimental results demonstrate that the proposed system can deliver reliable face recognition and effective suspicious-behavior detection within the constraints of a low-cost embedded platform.

Regarding face recognition, the system achieved 100% accuracy for owners, 95.56% for guests, and 95.23% for strangers. The perfect accuracy for owners highlights the stability of the enrollment procedure when sufficient training images are provided, while the slightly lower rates for guests and strangers reflect the impact of variability in facial expressions, illumination, and partial occlusions. These findings suggest that the LBPH algorithm, while efficient for embedded deployment, may benefit from further enhancement under more complex conditions. In particular, lightweight deep-learning-based methods can help maintain higher robustness without significantly increasing computational costs.

For suspicious-behavior detection, the module attained 95.10% detection accuracy with no false alarms, validating the effectiveness of combining dwell-time accumulation and repeated-appearance monitoring. The results indicate that the system successfully closes blind spots in conventional single-frame detection, such as strangers briefly exiting and re-entering to evade thresholds. The ability to dynamically register strangers ensured consistent recognition across noncontiguous appearances, thereby enabling more comprehensive behavioral monitoring.

Note also that the evaluation was conducted under controlled laboratory conditions with a limited number of participants. Large-scale deployments in diverse environments may introduce additional challenges, such as crowded scenes, variable lighting, and the simultaneous presence of multiple strangers. Furthermore, while the system demonstrated real-time performance with an average throughput of 12.87 FPS, future applications in high-traffic facilities may demand higher throughput and the ability to integrate with multi-camera networks.

The present experiments did not systematically vary facial expression, head angle, or occlusion/disguise (e.g., eyeglasses and sunglasses); robustness under these conditions remains to be quantified in future work. In addition, the camera-based module only monitors the primary entrance and does not address intrusion attempts through windows or other access points, which would require complementary sensors, such as window/door contact sensors, and is left for future integration. Regarding storage, two devices must be considered separately. The owner's

mobile device only stores the application itself and locally cached notifications or thumbnails, not the recognition database or video archive, so its storage requirement is negligible and well within the standard capacity of any modern smartphone. The Raspberry Pi's microSD card, by contrast, requires active management: temporary stranger profiles are discarded at the end of each session, and owner/guest enrollment is limited to 50 images per profile, so the storage footprint of facial feature data is on the order of a few tens of megabytes per registered user, while the recorded event video is the dominant consumer of microSD card space. As described in Sect. 3.4, the camera maintains a short-term rolling buffer (e.g., the preceding 5 min) that is continuously overwritten during normal operation and is committed to persistent storage as a fixed-length video clip only when an event is confirmed (e.g., a dwell-time or repeated-appearance alert), rather than through continuous 24 h recording; this ensures that each saved clip captures the behavior leading up to the triggering event. Because clips are written to the microSD card only upon confirmed alerts, storage consumption remains modest relative to conventional always-on surveillance systems; in our current deployment, a 64 GB microSD card is sufficient for this event-triggered usage pattern, and capacity can be expanded to 128 GB if a longer retention period or higher alert frequency is anticipated. Nonetheless, since microSD card capacity is finite and subject to write-cycle wear from repeated recording, a rolling local buffer with cloud offloading and a retention policy is recommended for continuous long-term operation.

In the current implementation, owner and guest profiles are enrolled once and remain unchanged thereafter, with no periodic re-enrollment or expiration mechanism. While this ensures a profile stays valid indefinitely once created, it makes the 50-image enrollment process relatively tedious for a guest who visits only infrequently. One possible direction for future work is an optional re-verification mechanism—for example, prompting a lightweight re-verification with fewer images (e.g., 10–15) if a guest profile has not been matched within a defined period (e.g., 30 days), rather than requiring full re-enrollment. This is one of several possible approaches to balancing enrollment convenience against recognition robustness and is left for future exploration.

The present prototype targets a single primary entrance of a residential unit or small office, with the camera mounted at approximately eye level facing the entry point. Extending the design to facilities with multiple points of entry—for example, separate units within a condominium—would require one camera-and-Raspberry-Pi node per entrance, coordinated through the mobile application and a shared backend database; this multi-camera, multi-entrance coordination is identified as future work in Sect. 6.

Overall, the discussion underscores that the proposed design is well suited for small- to medium-scale facilities, where real-time operation, cost-effectiveness, and reliable anomaly detection are critical. At the same time, the findings highlight the need for further optimization to extend applicability to more complex real-world environments.

6. Conclusion and Future Work

In this study, we presented the design and implementation of a smart access control system that integrates real-time face recognition with a suspicious-behavior detection module. By

employing dynamic enrollment and dual detection strategies—dwell-time accumulation and repeated-appearance monitoring—the system effectively addressed common evasion scenarios that conventional single-frame recognition approaches often overlook.

Experimental evaluation on a Raspberry Pi platform confirmed that the system can deliver high recognition accuracy (100% for owners, 95.56% for guests, and 95.23% for strangers), efficient processing performance (12.87 FPS with 0.39 s latency), and reliable suspicious-behavior detection (95.10% accuracy with no false alarms). These results validate the feasibility of deploying the proposed solution in resource-constrained embedded environments while maintaining practical usability and security.

Future work will focus on enhancing the robustness and scalability of the system. One direction involves exploring lightweight deep-learning architectures optimized for embedded platforms to improve recognition under challenging conditions such as illumination changes, facial occlusions, and diverse viewing angles. Another direction is the incorporation of advanced behavior analysis techniques, including multi-object tracking and optical-flow-based anomaly detection, to broaden the system's ability to capture complex suspicious activities. In addition, multi-camera coordination will be investigated to extend monitoring coverage to larger facilities. Finally, improvements to the human–machine interface, such as real-time voice intercom functions and more intuitive mobile app controls, will be pursued to strengthen user experience and emergency response capabilities.

Beyond access control, the underlying camera-based sensing and temporal behavior-analysis pipeline is, in principle, transferable to other identity- and behavior-aware monitoring scenarios, such as classroom engagement monitoring, eldercare fall and inactivity detection, or workplace safety-compliance checking. Exploring these generalizations is an interesting direction for future work.

In summary, the proposed system demonstrates a cost-effective and practical approach to intelligent access control, combining real-time recognition, behavioral monitoring, and multi-layered security mechanisms in a unified embedded framework.

Acknowledgments

The authors thank the reviewer for the valuable comments provided during the review process.

Author Contributions

N.-C. Chen was responsible for face recognition, suspicious-behavior detection, Raspberry Pi hardware integration, and electronic door-lock implementation. R.-Y. Chao was responsible for the mobile application implementation, including device–system connectivity, owner–guest interaction, and the post-event handling of suspicious activity. W.-F. Lu supervised the overall research design and methodology.

AI Assistance Disclosure

Under the authors' direction, Claude (Anthropic) was used to assist with language editing, reference verification, and manuscript formatting. All scientific content, data analysis, and conclusions are solely the work of the authors.

References

1 T. Ahonen, A. Hadid, and M. Pietikäinen: IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 2037. https://doi.org/10.1109/TPAMI.2006.244
2 Y. Taigman, M. Yang, M. Ranzato, and L. Wolf: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2014) 1701. https://doi.org/10.1109/CVPR.2014.220
3 F. Schroff, D. Kalenichenko, and J. Philbin: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2015) 815. https://doi.org/10.1109/CVPR.2015.7298682
4 J. Deng, J. Guo, N. Xue, and S. Zafeiriou: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2019) 4690. https://doi.org/10.1109/CVPR.2019.00482
5 K. Zhang, Z. Zhang, Z. Li, and Y. Qiao: IEEE Signal Process. Lett. **23** (2016) 1499. https://doi.org/10.1109/LSP.2016.2603342
6 W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld: ACM Comput. Surv. **35** (2003) 399. https://doi.org/10.1145/954339.954342
7 J. Núñez, Z. Li, S. Escalera, and K. Nasrollahi: Proc. 2024 IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops (WACVW) (2024) 251. https://doi.org/10.1109/WACVW60836.2024.00033
8 D. Helbing and P. Molnár: Phys. Rev. E **51** (1995) 4282. https://doi.org/10.1103/PhysRevE.51.4282
9 S. Ali and M. Shah: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2007) 1. https://doi.org/10.1109/CVPR.2007.382977
10 S. Yi, H. Li, and X. Wang: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (2015) 3488. https://doi.org/10.1109/CVPR.2015.7298971
11 M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette: Comput. Vis. Image Underst. **172** (2018) 88. https://doi.org/10.1016/j.cviu.2018.02.006
12 W. Luo, W. Liu, and S. Gao: Proc. IEEE Int. Conf. Comput. Vis. (2017) 341. https://doi.org/10.1109/ICCV.2017.45
13 R. Hinami, T. Mei, and S. Satoh: Proc. IEEE Int. Conf. Comput. Vis. (2017) 3619. https://doi.org/10.1109/ICCV.2017.391
14 Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh: IEEE Trans. Pattern Anal. Mach. Intell. **43** (2021) 172. https://doi.org/10.1109/TPAMI.2019.2929257
15 Raspberry Pi Foundation: https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf (accessed January 2025).