International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 15, No. 3, November 2026, pp. 746~755
ISSN: 2089-4864, DOI: 10.11591/ijres.v15.i3.pp746-755

Smart lock system with internet of things using facial recognition for home access

Patrick Ramos[1], José Echenique[1], Ricardo Yauri[2]
[1]Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima, Perú
[2]Facultad de Ingeniería, Universidad Tecnológica del Perú, Lima, Perú

Article Info

Article history:

Received Apr 14, 2026
Revised Jul 2, 2026
Accepted Jul 18, 2026

Keywords:

Automation
Biometric access
Computer vision
Facial authentication
Home security
Intrusion detection
Remote monitoring

ABSTRACT

Security in urban residential environments is a significant challenge in regions with high crime rates and unauthorized access. This has led to the integration of internet of things (IoT) technologies and biometric authentication mechanisms into access control systems. Research has described smart lock solutions that require significant hardware resources or computationally complex recognition algorithms. This research proposes the implementation of a low-cost distributed architecture for facial recognition, lock control, monitoring, and supervision using mobile devices. The proposed architecture performs facial recognition using a histogram-oriented gradient (HOG) model executed on a desktop processing node, while an ESP32 microcontroller activates the lock. Remote monitoring is achieved through a communication framework using Flask libraries, tunneling with Ngrok, and a mobile application. Experimental validation was performed in a controlled environment, achieving 96.2% accuracy, and an F1-score of 95.9%. Furthermore, the control subsystem achieved an activation time of 0.78 seconds. The results validate the integration of biometric authentication, IoT control, and remote monitoring. However, limitations were identified in low-light or spoofing environments.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Ricardo Yauri
Facultad de Ingeniería, Universidad Tecnológica del Perú
Cal. Natalio Sanchez Nro. 125, Lima, Perú
Email: boncer99@gmail.com

1. INTRODUCTION

Currently, security in urban homes is a critical issue due to burglaries and break-ins. In some countries, the rate of burglaries has increased in urban areas, making it necessary to use emerging technologies, such as the internet of things (IoT) [1] and artificial intelligence (AI) to improve home security systems [2]. In particular, the use of facial recognition algorithms in embedded devices [3] allows for automated access control, thus enhancing user safety [4]. However, locking mechanisms that use physical keys have limitations related to unauthorized access through manipulation with tools such as lock picks or cards, and the dependence on external services, which results in lost time [5]. Furthermore, hiring security personnel is an additional cost that increases the overall cost of home maintenance [6]. This highlights the limitations and the need to develop secure access control solutions [7], [8].

Research has focused on security and access control issues with electronic systems [9], [10]. For example, some systems use devices like Raspberry Pi [11] for facial recognition implementation and have efficiencies exceeding 90% [12], [13]. Solutions using sensors, radio frequency identification (RFID) antennas [14], or cameras for detecting and tracking people have also been investigated [15]. This indicates a research area for integrating embedded systems and machine learning.

In the systems developed in literature, the majority use of deep learning algorithms [16] or convolutional neural networks (CNNs) [17] is observed in facial recognition tasks due to their accuracy in the face of variations in lighting and capture angles [18]. Unlike traditional machine learning techniques, deep learning-based models detect complex patterns, with results exceeding 90% [10], [19], [20]. Furthermore, devices such as the Raspberry Pi can run these facial recognition algorithms with more complex architectures [21], unlike microcontrollers such as the Arduino UNO [22] or ESP32 [23].

In the field of home security systems, digital technologies have been used that are not suitable for controlling mechanical locks compared to intrusion techniques using facial recognition [5], [24], the use of ESP32-CAM [25], and electronic motion detection systems with PIR sensors [26]. IoT technologies applied to security and remote monitoring have been used for real-time control [27], [28], home monitoring with smartphones, cameras, and automatic notifications [29], [30] (Table 1).

Table 1. Comparison of related smart lock and facial recognition systems

References	Platform	Facial recognition	Connectivity	Remote monitoring	Mobile control
Pecolt *et al.* [3]	Raspberry Pi	Yes	Limited	No	No
Alzhrani *et al.* [4]	Smart Locker	Yes	Yes	Limited	No
Kolluru *et al.* [21]	Raspberry Pi	Si	Yes	Yes	No
PBV [24]	ESP32-CAM	AI face detection	Yes	Limited	No
Awangga [25]	ESP32-CAM	MTCNN	Yes	No	No
Proposed system	PC+ESP32	Histogram-oriented gradient (HOG)+face encoding	Yes	Yes (Flask+Ngrok)	Yes

Research has shown that it is possible to integrate facial recognition and IoT techniques for access control. However, most systems do not evaluate the practical integration of biometric authentication for residential environments. In addition, recognition systems using neural networks require computational resources, and although ESP32-based solution, they do not validate the interaction between facial recognition and IoT communication. This research proposes the development of an IoT smart lock system integrating facial recognition techniques to automate access to urban homes. The contribution of this work is not a new facial recognition algorithm but to show an integrated architecture with: i) HOG-based facial recognition, ii) control of locks with ESP32, iii) Flask and Ngrok tunnel services, and iv) control. From an embedded systems perspective, this work validates a hardware and software architecture where biometric processing is separated from real-time embedded control tasks.

2. RESEARCH METHOD

The system integrates a facial recognition stage with the IoT to automate access to a home. Operation begins when the user's face is captured by a 720-pixel camera, and a facial recognition algorithm developed in Python is executed on a computer. This algorithm incorporates a process based on deep neural networks that uses facial encodings [31], [32], and face detection is performed using the HOG method [33], [34]. HOG technique was selected in this work because the goal is to develop a low-cost and easy-to-implement process. Therefore, compared to CNN-based models, HOG requires fewer computational resources and has more restrictive hardware requirements. Furthermore, facial recognition employs a distance-based verification approach where each face has feature vectors that are compared to pre-stored encodings for identification. Subsequently, an ESP32 microcontroller receives the instruction to open or close the solenoid lock via a relay. Additionally, the algorithm's operation is observed through a mobile application, and the "tunneling" with Ngrok (Figure 1).

System processes are distributed between the desktop computer and the embedded system. Image acquisition, facial recognition using HOG, encoding generation, and identity verification are performed on the desktop computer due to hardware requirements. Meanwhile, the ESP32 microcontroller handles lock activation, relay control, status indication, and the execution of commands received from the recognition software. This task organization reduces the processing load on the embedded system while maintaining IoT-based control functionalities (Table 2). The architecture demonstrates a hardware-software co-design approach where a standard microcomputer handles resource-intensive facial recognition operations, while the ESP32 performs control tasks for lock activation.

2.1. Facial recognition algorithm design

To perform facial recognition, photos of an authorized user are stored for training the model using the HOG technique, and facial encodings are generated in a "pickle" file. The video feed containing the detection is transmitted via the Flask framework to a local web server. The flowchart illustrates the registration process

for a new authorized user. The process begins with the user entering their name. The data is then stored in folders, and when the camera becomes available, a loop starts to display the images in real time. The user interacts with the system to take a picture of the frame and generate a file with the date and time stamp (Figure 2).

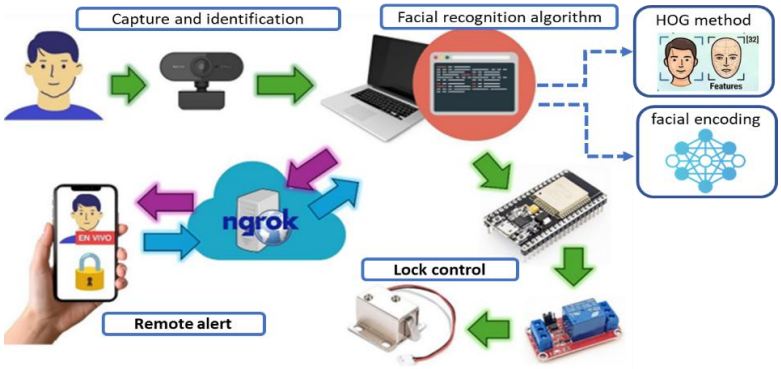


Figure 1. Project component diagram

Table 2. Functional distribution of tasks within the proposed architecture

Component	Main function
Desktop computer	Face detection using HOG, facial encoding, identity verification, and video streaming
ESP32	Relay activation, lock actuation, LED status indication, and command execution
Flask server	Transmission of video stream and recognition results
Ngrok service	Secure remote access to local monitoring server
Mobile application	Remote monitoring and manual lock control

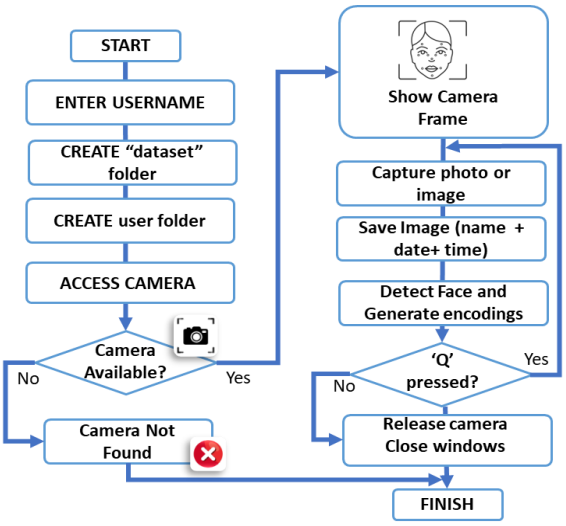


Figure 2. Flowchart for registering a new user

The programming class called "Config" allows for storing global configuration and contains the attribute "PERSON_NAME". The "FolderManager" class manages the folders where images are stored and the content of the attribute used to construct file system paths ("name" in Spanish "Nombre", "dataset_folder", and "persona_folder"). The main class TakePhoto encapsulates the photo capture process, the attributes of the username, the instance present in "FolderManager", the camera access variable ("cap"), and the counter of photos taken (Figure 3). The image is then loaded for blue, green, red (BGR) to red, green, blue (RGB) format conversion, faces are located using the HOG technique.

Experimental validation was performed in a controlled environment using a webcam with a resolution of 1280×720 pixels. This camera was used to create the dataset with images of users selected during the

acquisition tests. For each authorized user, facial images were captured with different head orientations. Performance was measured using the metrics of accuracy, completeness, and F1-score obtained from the confusion matrix. For identity verification, the Euclidean distance between the facial encoding and the recorded encodings was calculated (Figure 4).

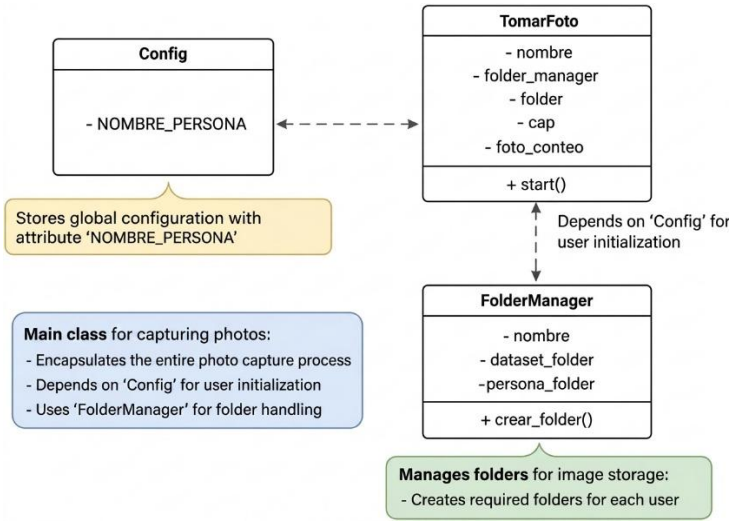


Figure 3. Class diagram for user registration

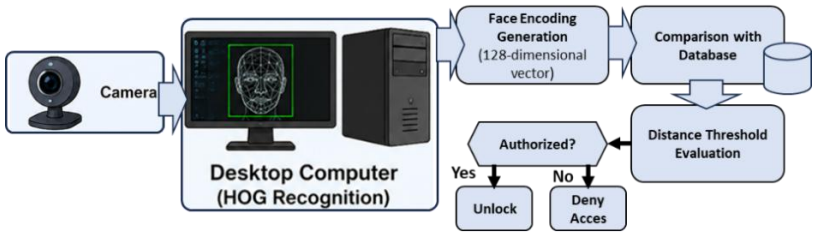


Figure 4. Facial recognition workflow

2.2. ESP32-based electronic device for solenoid lock

To use the system, the user stands in front of the camera to capture their image. If the face matches that of an authorized user, the command "ON" is sent to the ESP32 via the serial port, indicating that the lock should be activated. Upon receiving this command, the ESP32 activates pin 12, allowing the lock to open, and pin 2 indicates that the lock is open. For the next 5 seconds, the system continuously monitors the situation, and then the lock pin and the warning LED are automatically deactivated (Figure 5). The ESP32-WROOM-32D module is powered via a universal serial bus (USB) connection to the desktop computer that runs the Python code for face classification (Figure 6). In the monitoring process, the application serves as a supervisory interface where users access the live video feed, verify the results, and remotely generate lock control commands.

2.3. User interface for facial recognition

An Android application was designed using MIT App Inventor to display captured video and labels for detected subjects (Figure 7). This remote monitoring is restricted to use within the same Wi-Fi network as the desktop computer. Therefore, the Ngrok service is used as an intermediary to access the local server with the Flask web application from outside the local network. The smartphone accesses this public URL, Ngrok receives the request and redirects it through the secure tunnel to the desktop computer. Flask then processes the request and responds.

The app is built using conditional programming blocks that control the lock and data transmission. At the top is a box displaying the video stream, linked to the ngrok address. Below are buttons to refresh the page and another to manually lock the screen for 10 seconds. Button 2 allows manual control of the lock. Additionally, Button 3 allows refreshing the stream with a "Reload" command (Figure 8).

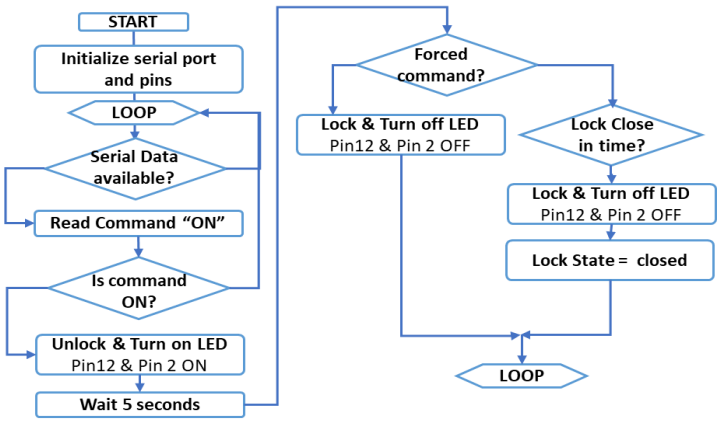


Figure 5. Programming flowchart for the ESP32

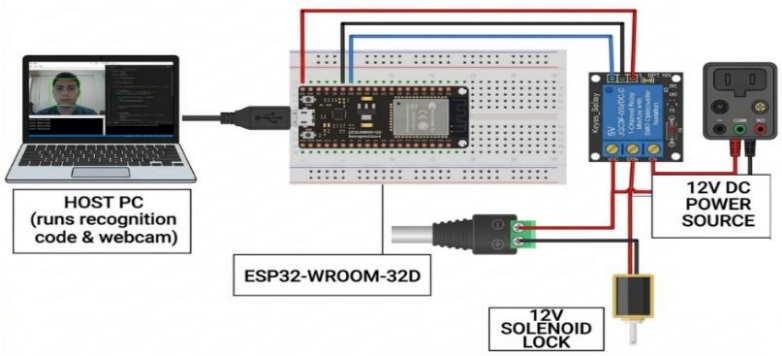


Figure 6. Circuit interconnection diagram

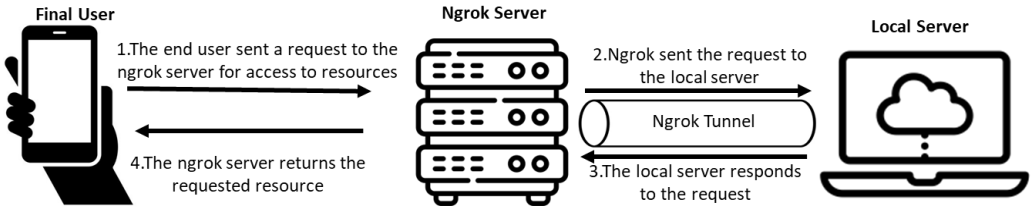


Figure 7. Remote monitoring architecture using Flask, Ngrok tunneling service and mobile application

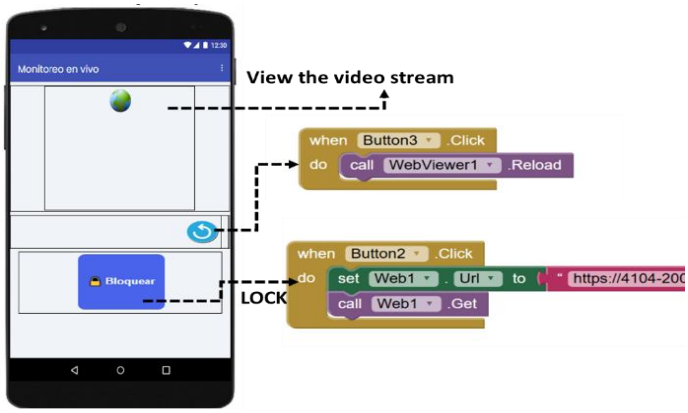


Figure 8. App design in MIT App Inventor

3. RESULTS AND DISCUSSION

3.1. Facial recognition

The facial recognition stage enabled the detection of faces under different lighting conditions, postures, and with the addition of objects to the face (such as glasses). Furthermore, the gradient map of the HOG method, showing a combination of edges and orientations, is displayed below each photo. The images stored in the folder are in RGB format, while the facial encodings are 128-dimensional numerical vectors representing the distance between the eyes, jaw shape, and proportions. These do not contain information concerning pixels or the physical shape of the face (Figure 9).

On the other hand, after the model is trained, approved faces are visually displayed with a label within a 1-meter range. One drawback is the detection distance using the "face_distance" function, so the best match is selected using the "argmin" function. This ensures that only the most similar face is accepted, reducing false positives. The algorithm is still susceptible to light covering part of the face due to the HOG method (Figure 10). Regarding the user “patrick”, facial recognition was not achieved for photos with strong contrast between light and dark areas. Therefore, these results indicate that the algorithm has difficulty recognizing registered users, as evidenced by the recall value and false negatives. For this reason, it was decided to reduce the sensitivity of facial recognition by using a maximum Euclidean distance threshold between the face encoding and the camera (Table 3).

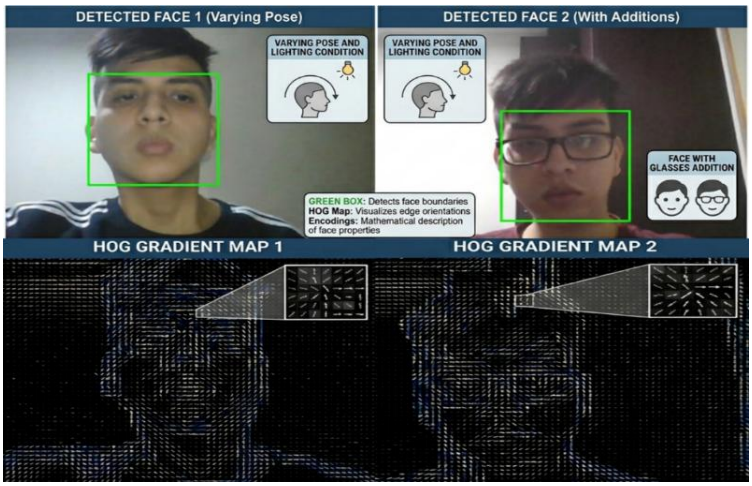


Figure 9. Face detection using the HOG method

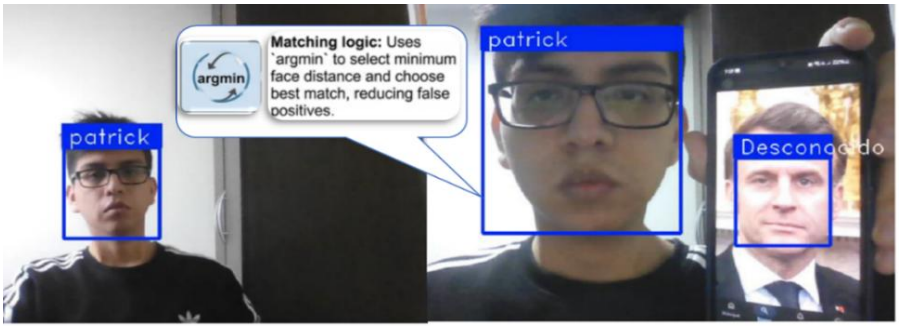


Figure 10. Recognition of authorized and unauthorized users

Table 3. Performance parameters for the facial recognition algorithm

Parameter	Value	Formula	Interpretation
Accuracy	96.2%	TP+TN+FP+FNTP+TN	High overall recognition performance
Precision	100%	TP+FPTP	No unauthorized access accepted
Recall	92.2%	TP+FNTP	Some authorized users not recognized
F1-score	95.9%	2∗precision+recallprecision∗recall	Balanced recognition performance
time	0.78 s	Average unlocking time	Near real-time operation

TP is true positive, TN is true negative, FP is false positive, and FN is false negative

The difference between accuracy and the lower exhaustiveness of 3.8% indicates that recognition errors are related to false rejections. During testing, some authorized users were not recognized in environments with contrasting lighting. The 96.2% accuracy indicates that the system architecture provides adequate biometric authentication for home access control applications. Furthermore, the perfect accuracy (100%) indicates that no unauthorized access was granted during testing, while the completeness (92.2%) suggests that errors are due to false rejections affecting authorized users. On the other hand, compared to facial recognition approaches using deep learning [16]-[18], the use of HOG techniques requires fewer processing resources on low-cost devices. In addition, the unlock time of 0.78 s shows that the integration of facial recognition, communication, and lock activation has a delay imperceptible to users, demonstrating that the system's task distribution allows the ESP32 to focus on actuation, while the recognition processes are executed on the desktop platform device.

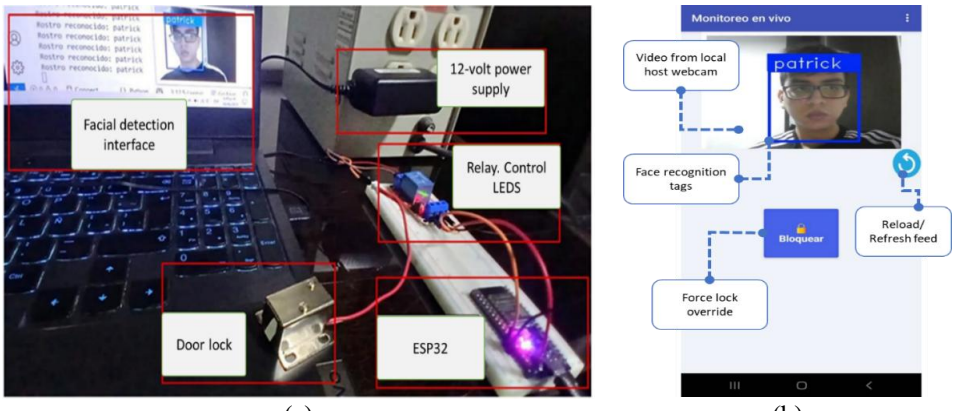
3.2. ESP32 for interconnection with the camera and solenoid lock

The circuit components were integrated and connected using the ESP32-WROOM-32D module, a relay module with an optocoupler, and a 12 V solenoid lock. The connection to the ESP32 was validated by developing a Python program on a computer to run the facial recognition algorithm and verify access from the mobile application (Figure 11). When an authorized user's face is recognized, an activation signal is sent from the ESP32 to activate the electronic lock (Figure 11(a)). Furthermore, variations between measurements are due to differences in image acquisition time, processing latency, and potential mechanical errors. The measured response time of 0.78 s corresponds to the cumulative delay generated by the stages of image acquisition, facial detection, facial encoding, identity verification, serial communication, and lock activation.

3.3. Monitoring with the facial recognition system

In addition to facial authentication, the monitoring stage allows for real-time viewing of access events via a mobile application. In this case, the user remotely observes the video feed, verifying the identity tags assigned by the recognition algorithm and manually forcing the lock closed. This stage contributes to the automated authentication process with human oversight. Unlike conventional electronic lock systems that only perform authentication, this system integrates remote visual monitoring with a mobile device, allowing access controls to not depend solely on the facial recognition algorithm. Because the Flask framework is used, it's only possible to monitor the camera's view when the smartphone is on the same local network as the computer running the detection model. Ngrok's tunneling functionality allows access from external environments (Figure 11(b)).

Furthermore, the execution time for the processes was obtained during the 10 access tests. The system takes 0.87 and 0.66 seconds to open and close the lock, respectively. Likewise, the longest and shortest durations with the lock open were 5.72 and 5.39 seconds, respectively (Table 4).



(a) (b)

Figure 11. Validation of the control program's functionality with ESP32 using; (a) implemented physical circuit and (b) remote monitoring with the mobile app

3.4. Security considerations and limitations

Similar to other access control systems, this architecture is exposed to cybersecurity risks such as unauthorized access, remote monitoring interfaces, credential theft, and denial-of-service attacks. Furthermore, the facial recognition stage is vulnerable to photo spoofing attacks because it lacks a liveness detection

mechanism, which is beyond the scope of this project. While Ngrok has secure tunneling capabilities, a comprehensive cybersecurity assessment is beyond the scope of this research. Therefore, future work could incorporate liveness detection, encrypted communication channels.

On the other hand, facial recognition processing is currently performed on a standard microcomputer. The system architecture shows how computational and control tasks can be distributed across different platforms. This approach is suitable for current IoT systems, where devices with limited hardware resources work in conjunction with other nodes that have greater processing capacity.

Table 4. Time until the lock opens and opening time in seconds

Attempt number	Time until opening (seconds)	Opening duration (seconds)
1	00.86	05.52
2	00.87	05.47
3	00.73	05.72
4	00.80	05.39
5	00.67	05.51
6	00.80	05.43
7	00.66	05.42
8	00.87	05.39
9	00.66	05.71
10	00.80	05.46

4. CONCLUSION

A smart lock system with the IoT and a facial recognition algorithm was implemented for home access using a face detection model that compares faces with encodings. The solenoid lock was efficiently controlled with an ON-OFF relay module and an ESP32 microcontroller, simulating the conditions of a video surveillance camera.

Furthermore, the Flask framework was used to transmit the algorithm's response and the physical lock's status. To design the user interface, the mobile application was integrated with remote access via secure tunnels using the Ngrok service. The remote monitoring function allows access control from external networks via a secure tunnel architecture. This creates an additional layer of security, as lock control is complemented by human verification and intervention. The algorithm performed adequately for identifying authorized users. Furthermore, the F1-score performance parameter showed a value of 95.9%, similar to the values for “accuracy” (96.2%), “precision” (100%), and “recall” (92.2%).

The mobile app interface validated the remote lock-forcing functionality. Additionally, the image refresh button was implemented to address potential connectivity issues. Furthermore, the facial recognition algorithm was unable to distinguish a photograph from a real face, so the authorized user of the mobile app can restrict access for a specific period of time.

During system validation, a limitation was identified related to facial recognition performance, which is reduced by lighting contrasts and facial occlusion caused by the HOG technique. Furthermore, the current implementation does not integrate mechanisms to prevent access using photographs. Finally, the validation was performed under controlled conditions with a limited number of users.

The main scientific contribution is the validation of a low-cost, integrated architecture with biometric authentication, IoT-based lock control, and mobile monitoring within a residential security framework. Furthermore, instead of developing a new recognition algorithm, the feasibility of integrating technologies into a single integrated security solution is demonstrated. The current architecture is based on the use of a conventional computing system with the facial recognition stage that can be adapted to reduced hardware devices, allowing its portability and future migration to edge AI solutions to be extrapolated.

Future research could integrate biometric validation stages to prevent spoofing (such as liveness detection and blink detection). Additionally, deep learning models (e.g., TensorFlow Lite) could be evaluated for facial recognition. Multimodal authentication mechanisms integrating RFID or fingerprint technologies could also be implemented to enhance security.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Patrick Ramos	✓	✓	✓	✓		✓		✓	✓	✓				
José Echenique	✓	✓	✓	✓	✓	✓		✓	✓					
Ricardo Yauri	✓		✓	✓						✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author R.Y upon reasonable request.

REFERENCES

[1] M. S. Amine, F. A. Nada, and K. M. Hosny, "Improved model for intrusion detection in the Internet of Things," *Sci. Rep.*, vol. 15, no. 1, p. 21547, 2025, doi: 10.1038/s41598-025-92852-6.

[2] H. A. S. Ali and V. R. J., "Deep Learning Approach for Protecting IoT Smart Home Devices Against Multiclass Botnet Attacks," *Int. J. Comput. Networks Appl.*, vol. 12, no. 3, pp. 307–323, Jun. 2025, doi: 10.22247/ijcna/2025/20.

[3] S. Pecolt, A. Błażejewski, T. Królikowski, I. Maciejewski, K. Gierula, and S. Glowinski, "Personal Identification Using Embedded Raspberry Pi-Based Face Recognition Systems," *Appl. Sci.*, vol. 15, no. 2, pp. 1-27, 2025, doi: 10.3390/app15020887.

[4] A. A. Alzhrani, M. Balfaqih, F. Alsenani, M. Alharthi, A. Alshehri, and Z. Balfagih, "Design and Implementation of an IoT-Integrated Smart Locker System utilizing Facial Recognition Technology," *Eng. Technol. Appl. Sci. Res.*, vol. 14, no. 4, pp. 16000–16010, Aug. 2024, doi: 10.48084/etasr.7737.

[5] H. Putri *et al.*, "Security System for Door Locks Using YOLO-Based Face Recognition," *JOIV Int. J. Informatics Vis.*, vol. 9, no. 1, pp. 224-230, Jan. 2025, doi: 10.62527/joiv.9.1.2410.

[6] M.-T. Yeh, Y.-C. Tsai, C.-H. Cheng, Y.-N. Chung, and P.-S. Lu, "Smart Home Surveillance System Based on the Optimized EfficientDet Network," *Sens. Mater.*, vol. 35, no. 7, pp. 2567–2579, Jul. 2023, doi: 10.18494/SAM4297.

[7] N. Shashikala and M. R. Mundada, "Internet of Things (IoT) for Secure Data and M2M Communications—A Study," in *Lect. Notes Data Eng. Commun. Technol.*, vol. 139, pp. 13–28, 2023, doi: 10.1007/978-981-19-3015-7_2.

[8] N. A. M. Razali, F. S. Abdullah, N. S. M. Noorpi, N. A. A. Talib, S. Mohammad, and T. N. Zaidi, "IoT-based home security: A facial recognition approach," in *AIP Conf. Proc.*, Oct. 2025, doi: 10.1063/5.0298762.

[9] Y. Akbari, A. Al-Binali, A. Al-Mohannadi, N. Al-Hemaidi, O. Elharrouss, and S. Al-Maadeed, "A New Framework for Smart Doors Using mmWave Radar and Camera-Based Face Detection and Recognition Techniques," *Sensors*, vol. 24, no. 1, pp. 1-18, Dec. 2023, doi: 10.3390/s24010172.

[10] M. Kumar, K. S. Raju, D. Kumar, N. Goyal, S. Verma, and A. Singh, "An efficient framework using visual recognition for IoT based smart city surveillance," *Multimed. Tools Appl.*, vol. 80, no. 20, pp. 31277–31295, 2021, doi: 10.1007/s11042-020-10471-x.

[11] S. B. Chaabane, R. Harrabi, A. Bushnag, and H. Seddik, "Development of Surveillance Robots Based on Face Recognition Using High-Order Statistical Features and Evidence Theory," *J. Imaging*, vol. 12, no. 3, pp. 1-21, 2026, doi: 10.3390/jimaging12030107.

[12] I. P. A. Bayupati, A. Ersapramana, and I. P. A. Dharmaadi, "Design and Build of Masked Face Identification System and IoT-Based Body Temperature Measurement," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 13, no. 4, pp. 1582–1588, Aug. 2023, doi: 10.18517/ijaseit.13.4.17917.

[13] B. Bhuckory and S. Pudaruth, "A Multimodal IoT-Based Home Intrusion Detection System," *Int. J. Comput. Theory Eng.*, vol. 15, no. 3, pp. 117–124, 2023, doi: 10.7763/IJCTE.2023.V15.1340.

[14] E. Abdullahu, H. Wache, and M. Piangerelli, "Secure and Decentralized Hybrid Multi-Face Recognition for IoT Applications," *Sensors*, vol. 25, no. 18, pp. 1-17, Sep. 2025, doi: 10.3390/s25185880.

[15] O. C. S. and O. O. M., "Development of a Facenet Enhanced Secured Smart Office System," *Int. J. Intell. Syst. Appl.*, vol. 16, no. 1, pp. 41–51, Feb. 2024, doi: 10.5815/ijisa.2024.01.04.

[16] S. B. Chaabane, R. Harrabi, and H. Seddik, "Home security system with hybrid face recognition approach using convolutional neural Networks, fuzzy Logic, and SVM classifier," *Multimed. Tools Appl.*, vol. 85, no. 2, 2026, doi: 10.1007/s11042-026-21261-2.

[17] L. J. Yang and Y. Li, "A Deep Learning-Based Framework for Robust Facial Keypoint Localization in Unconstrained Conditions," *Teh. Vjesn. - Tech. Gaz.*, vol. 33, no. 1, pp. 258–270, Feb. 2026, doi: 10.17559/TV-20240824001940.

[18] T. L. Liao, H. C. Chen, Q. H. Song, and Y. Y. Hou, "Face recognition and real-time tracking system based on convolutional neural network and parallel-cascade PID controller," *Meas. Control (United Kingdom)*, vol. 55, no. 7–8, pp. 616–630, Jul. 2022, doi: 10.1177/00202940221089237.

[19] M. Marimuthu, G. Mohanraj, J. Akilandeswari, and V. Sathiyapriya, "Facial Recognition Enabled Smart Security Lock System Using Machine Learning Approach," *EAI Endorsed Trans. Internet Things*, vol. 11, no. 1, p. 10, Jun. 2025, doi: 10.4108/eetiot.5657.

[20] M. A. S. Ali, M. M. Eltoukhy, F. Rajeena P. P., and T. Gaber, "Efficient thermal face recognition method using optimized curvelet features for biometric authentication," *PLoS One*, vol. 18, no. 6, pp. 1-33, Jun. 2023, doi: 10.1371/journal.pone.0287349.

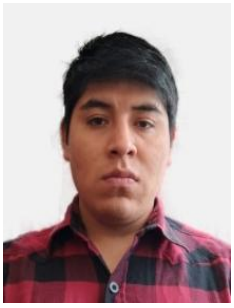
[21] C. Kolluru, G. V. Akhil, S. S. Priyanka, D. K. Reddy, and A. S. Kumar, "Development of Face Recognition-Based Smart Door Lock System with Remote Servo Control Authentication," in *IEEE Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT),*

ICCCNT 2023, Delhi, India: IEEE, 2023, pp. 1-6, doi: 10.1109/ICCCNT56998.2023.10307437.
[22] F. N. Lattab, B. Belaid, G. Belalem, and A. Bouamar, "Enhanced Home Safety Through Intelligent Monitoring and Alerting," *Lect. Notes Netw. Syst.*, vol. 1796, pp. 1–12, 2026, doi: 10.1007/978-3-032-15407-1_1.
[23] R. K. Satyanarayana, R. G. S. N., M. Murali, and M. Satyanarayana, "Double Loop Antenna Design For Smart Door Lock System Using Iot Applications," *Salud Cienc. Tecnol. - Ser. Conf.*, vol. 3, no. 1, p. 10, Aug. 2024, doi: 10.56294/sctconf2024.1125.
[24] R. R. PBV, V. S. Mandapati, S. L. Pilli, P. L. Manojna, T. H. Chandana, and V. Hemalatha, "Home Security with IOT and ESP32 Cam - AI Thinker Module," in *Int. Conf. Cogn. Robot. Intell. Syst. (ICC-ROBINS)*, IEEE, Apr. 2024, pp. 710–714, doi: 10.1109/ICC-ROBINS60238.2024.10533960.
[25] R. A. Awangga, A. Faroqi, E. A. Z. Hamidi, P. P. N. Chidayah, D. Eryanti, and S. R. Muslimah, "Design of a Prototype Face Recognition System for Home Security Based on Multi-Task Convolutional Neural Network (MTCNN) Using ESP32-CAM," in *2025 11th Inte. Conf. on Wireless and Telematics (ICWT)*, Jul. 2025, pp. 1–6, doi: 10.1109/ICWT66752.2025.11181924.
[26] S. Madura, S. Janani, and V. R. Kanagavalli, "Smart Home Security System with Integrated Motion Detection and Facial Recognition," in *2024 Int. Conf. Smart Technol. Sustain. Dev. Goals (ICSTSDG)*, IEEE, Nov. 2024, pp. 1–6, doi: 10.1109/ICSTSDG61998.2024.11026660.
[27] N. F. Awang, A. F. I. M. Zainudin, S. Marzukhi, N. A. A. Jamaludin, and S. N. A. S. Zakaria, "Replay Attack on Bluetooth Communication with Software Defined Radio in the IoT-Based Smart Home," *JOIV Int. J. Informatics Vis.*, vol. 10, no. 1, pp. 448-453, Jan. 2026, doi: 10.62527/joiv.10.1.2588.
[28] M. A. Addow, A. D. Jimale, A. Y. Nageye, M. O. Abdullahi, and S. M. Hilowle, "A low-cost IoT-based smart home automation system for urban sustainability in Mogadishu, Somalia," *Discov. Int. Things*, vol. 5, no. 1, pp. 1-18, 2025, doi: 10.1007/s43926-025-00207-4.
[29] G. Parween, A. Al-Anbuky, G. Mawston, and A. Lowe, "Digital Twin Prospects in IoT-Based Human Movement Monitoring Model," *Sensors*, vol. 25, no. 21, p. 6674, Nov. 2025, doi: 10.3390/s25216674.
[30] M. F. Gabriel *et al*., "Smartphone-enabled real-time data on home electricity use and indoor air quality to promote energy-saving behaviors," *Energy Build.*, vol. 347, no. 1, pp. 1-12, Nov. 2025, doi: 10.1016/j.enbuild.2025.116344.
[31] J. P. Lee *et al.*, "Encoding of multi-modal emotional information via personalized skin-integrated wireless facial interface," *Nat. Commun.*, vol. 15, no. 1, pp. 1-13, Jan. 2024, doi: 10.1038/s41467-023-44673-2.
[32] M. Stettler, A. Lappe, and M. A. Giese, "Facial expression recognition based on multi-domain norm-referenced encoding," *Neural Netw.*, vol. 197, no. 1, pp. 1-13, May 2026, doi: 10.1016/j.neunet.2025.108384.
[33] S. S. F. A. Elnozahy, S. C. Pari, and L. C. Liang, "Raspberry Pi-Based Face Recognition Door Lock System," *IoT*, vol. 6, no. 2, pp. 1-22, May 2025, doi: 10.3390/iot6020031.
[34] C.-B. Yao and C.-T. Lu, "Dynamic Tracking and Real-Time Fall Detection Based on Intelligent Image Analysis with Convolutional Neural Network," *Sensors*, vol. 24, no. 23, pp. 1-25, Nov. 2024, doi: 10.3390/s24237448.

BIOGRAPHIES OF AUTHORS



Patrick Ramos is a graduate of the Faculty of Electronic and Electrical Engineering at the Universidad Nacional Mayor de San Marcos. During his studies, he specialized in the design of embedded systems and the control of electronic devices. His contribution consisted of implementing hardware based on the ESP32, managing the activation of the solenoid lock and providing visual feedback through LED indicators. He has a strong interest in the development of internet of things (IoT). His areas of expertise include C++ programming, the management of optocoupled relays, and the integration of 12 V DC power supplies. He can be contacted at email: patrick.ramos@unmsm.edu.pe and patrickrv72@gmail.com.



José Echenique is a graduate of the Faculty of Electronic and Electrical Engineering at the Universidad Nacional Mayor de San Marcos. He has specialized in software development for computer vision and web application architecture for remote monitoring. In the project, he was responsible for implementing the facial recognition algorithm using the HOG method and generating 128-dimensional encodings for authorizing users. His technological interests focus on Flask server deployment, network tunneling using Ngrok, and image processing optimization in Python. He can be contacted at email: jose.echenique@unmsm.edu.pe and joan.ecyu@gmail.com.



Ricardo Yauri is a Master of Science in Electronic Engineering with a Mention in Biomedical. He is associate professor at the Universidad Nacional Mayor de San Marcos (UNMSM), Universidad Tecnológica del Perú (UTP) and Ph.D. student in Systems Engineering. He is professor at Universidad Tecnológica del Perú and Universidad Peruana del Norte. He has participated as a teacher in courses oriented to the Internet of things and applications in home automation and the Cisco academy for IoT. He was a researcher at INICTEL-UNI in the Embedded Systems and internet of things research group. He developed research projects on the implementation of low consumption IoT devices that involve inference techniques, and machine learning algorithms. He can be contacted at email: c24068@utp.edu.pe, boncer99@gmail.com, ryauri@untels.edu.pe, and ryaurir@unmsm.edu.pe.