UDC 004.89

DESIGN AND IMPLEMENTATION OF AN IOT-BASED ACCESS CONTROL SYSTEM COMBINING AUTOMATIC LICENSE-PLATE RECOGNITION AND RADIO-FREQUENCY IDENTIFICATION TECHNOLOGIES

Nazar Omeliukh*, *Halyna Klym**, *Taras Tkachuk
Lviv Polytechnic National University,
12 Stepan Bandera St., Lviv 79013, Ukraine

Omeliukh, N.S., Klym, H.I., Tkachuk, T.I. (2025). Design and Implementation of an IoT-Based Access Control System Combining Automatic License-plate Recognition and Radio-frequency Identification Technologies. *Electronics and Information Technologies*, 32, 131–140. https://doi.org/10.30970/eli.32.9

ABSTRACT

Background. Ensuring security for private areas and infrastructure hubs is a growing concern in the modern world. Traditional methods, such as human guards and mechanical barriers operated by physical tokens (keys or cards), are often slow, inefficient, and prone to security risks like unauthorized duplication or theft. Furthermore, legacy systems lack comprehensive auditing capabilities. This creates a critical need for modern, automated IoT-based systems that ensure reliable access management and real-time monitoring.

Materials and Methods. The system uses several electronic components. The core is a low-cost microcontroller with a camera module. A radio-frequency identification (RFID) reader scans access cards. An ultrasonic distance sensor detects obstacles for safety, and a servo motor operates the physical barrier. The software backend was developed in Python, with a JavaScript (React) web control panel.

The system combines two identification methods. First, a camera captures a vehicle's image, sending it to a server where an AI model reads the license plate. The server checks the number against an approved list. If not recognized, the driver scans an RFID card as a secondary method. A distance sensor continuously monitors the barrier area to prevent closing on an obstacle. A web interface allows an operator to monitor the camera, review logs, and manually open the barrier.

Results. The developed system was tested successfully. The AI model achieved 75% accuracy in identifying license plates. The system proved fast, with an average response time from image capture to decision under one second. The safety sensor was validated, reliably detecting obstacles and preventing barrier movement, ensuring safe operation. Provided results of comparing video quality and system response time. The optimal balance between video quality and speed was found at 800x600 resolution.

Conclusions. A reliable, cost-effective automated access control system was successfully designed, built, and tested. The combination of AI-based license plate reading with a backup RFID system provides a robust, flexible solution. This system is well-suited for improving security and efficiency in real-world applications like residential, office, and industrial zones.

Keywords: automated access control, ESP32-S3, ALPR, RFID, security.

INTRODUCTION

In the modern world, the challenge of ensuring security and managing access to private territories, corporate facilities, and infrastructure hubs is becoming increasingly critical. The necessity for implementing robust automated systems has grown significantly, driven by constantly strengthening security requirements and a desire to minimize the risks of unauthorized access [1]. Traditional control methods, which served as a basic level of protection for decades, are now demonstrating their inefficiency in a dynamic environment. Systems that rely exclusively on mechanical barriers and physical tokens like keys or cards suffer from significant vulnerabilities, including the risk of loss, theft, or unauthorized duplication [1][2]. Furthermore, these legacy systems inherently lack the capabilities for comprehensive auditing or real-time event monitoring, making it difficult to maintain a reliable record of who accessed an area and when [1].

In contrast to outdated approaches, the modern security paradigm actively integrates Internet of Things (IoT) technologies, artificial intelligence (AI), and cloud computing. This integration transforms access control systems from simple locking mechanisms into intelligent ecosystems capable of data collection, analytics, and autonomous decision-making [2]–[4]. Modern automated access control systems are becoming part of a broader "smart building" or "smart city" infrastructure, where access data can be used for optimizing operations, managing personnel, and proactively identifying security threats [2]. The development of embedded systems and microelectronics has enabled the creation of more reliable, cost-effective, and autonomous solutions.

A key role in such advanced systems is played by computer vision, particularly as it applies to deep learning models running on peripheral devices. The capability to deploy sophisticated algorithms directly "at the edge" is crucial for reducing latency and enhancing privacy [5]–[6]. This study utilizes a powerful microcontroller (ESP32-S3) capable of handling such edge-computing tasks, specifically for processing video streams for Automatic License Plate Recognition (ALPR). The effectiveness of ALPR is highly dependent on the underlying detection model. The integration of modern, lightweight object detection models like YOLOv8 allows for high accuracy and processing speed in real-time recognition tasks, which is critical for a responsive access system [7].

However, to create a truly flexible and reliable system, a single authentication method is insufficient [3]. This study presents the architecture of a comprehensive access control system that integrates multiple technologies: ALPR as the primary contactless method, radio-frequency identification (RFID) as a secondary verification layer, and remote management via a web interface. To manage these components, a robust client-server architecture is adopted. A lightweight backend service built with Python Flask handles data processing and business logic, while a separate React-based frontend provides a dynamic interface for visualization and user interaction [6], [8].

This work contributes to the field by presenting a detailed, practical, and end-to-end implementation of such an integrated system. The deliberate combination of ALPR and RFID authentication creates a robust, multi-layered security approach [3]. This dual-modal design ensures high reliability [4], allowing RFID to serve as a dependable fallback mechanism in scenarios where ALPR might be compromised by adverse weather, poor lighting, or obscured license plates [2], [6]. Furthermore, the system design incorporates essential operational components, such as distance sensors for barrier safety, demonstrating a holistic approach that considers not only security but also practical usability. The primary aim of this paper is therefore to present the complete system architecture, detail the specific hardware solution, and validate its performance, particularly the high efficiency of the ALPR module in real-world conditions.

HARDWARE AND SOFTWARE IMPLEMENTATION

The system is designed to automatically recognize vehicles using ALPR, support RFID-based access as secondary authentication, ensure safe barrier operation via a distance sensor, transmit data wirelessly between the microcontroller and the backend server, and provide a web interface for monitoring and remote control. A well-defined architecture ensures smooth integration of hardware and software components. The system comprises a physical access barrier, embedded processing on the ESP32-S3, a backend service, and a user-friendly web interface. The system architecture is shown on **Fig. 1**.

This access control system is built on a modular, distributed architecture, ensuring ease of use, high component integration, and reliable operation. The overall structural diagram illustrates interactions between the main components: the user (operator), the web interface, the server, and the ESP32-S3 microcontroller.

The ESP32-S3 microcontroller is used as the primary control unit due to its processing power, on-board Wi-Fi, and rich GPIO set [9]. ESP32-S3 handles communication and access control; collects inputs from the camera, RFID reader, and distance sensor, and executes decisions received from the backend. System designed in two-way directions for entry and exit. For each direction, we used one microcontroller and the proper modules.

OV3660 camera captures images for ALPR. On the ESP32-S3 CAM board, the camera connects via the integrated camera connector [10]. The camera is configured through the driver with optimized parameters (clock, frame size, pixel format, brightness/mirroring/saturation) for stable streaming. PN532 RFID module provides alternative authentication via RFID cards [11]. It is connected over I2C on the ESP32-S3, enabling reliable tag reading in proximity access scenarios. HC-SR04 distance sensor prevents accidental barrier operation by detecting vehicles or pedestrians near the barrier [12]. The servo motor drives the physical barrier mechanism (PWM control from an ESP32-S3 GPIO). The servo is selected for precision and reliability under repetitive motion. The system architecture with selected hardware is shown on **Fig. 2.**

Firmware (ESP32-S3) developed using PlatformIO. The boot flow initializes Wi-Fi and peripheral modules, starts the embedded web server, and enters the main loop for event handling. The firmware ensures automatic Wi-Fi reconnection to maintain continuous service availability. Also integrated a Wi-Fi manager that provides user friendly network connection. Firmware based on an asynchronous web server.

The ESP32-S3 acts as the core processing unit. It acquires video frames from the OV3660 camera, polls the PN532 for RFID tags when present, reads the HC-SR04 to prevent hazardous barrier motion, and exchanges data and commands with the backend over Wi-Fi.

The firmware follows a deterministic initialization and event-driven loop. Firstly, the firmware initializes storage and connectivity: mounts SPIFFS for static assets, configures Wi-Fi in AP or STA mode, and ensures automatic reconnection to maintain continuous availability. Then the firmware initializes peripherals: configure the OV3660 camera (frame size, pixel format and stability parameters), set up the PN532 over I2C for RFID reads, initialize the HC-SR04 for obstacle detection, and prepare PWM control for the servo.

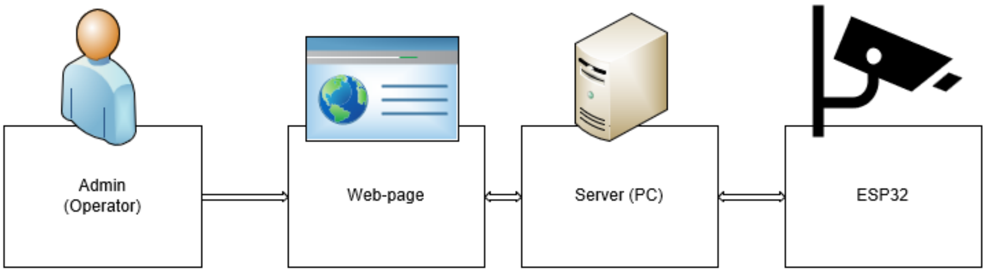


Fig. 1. System architecture, showing interactions between components.

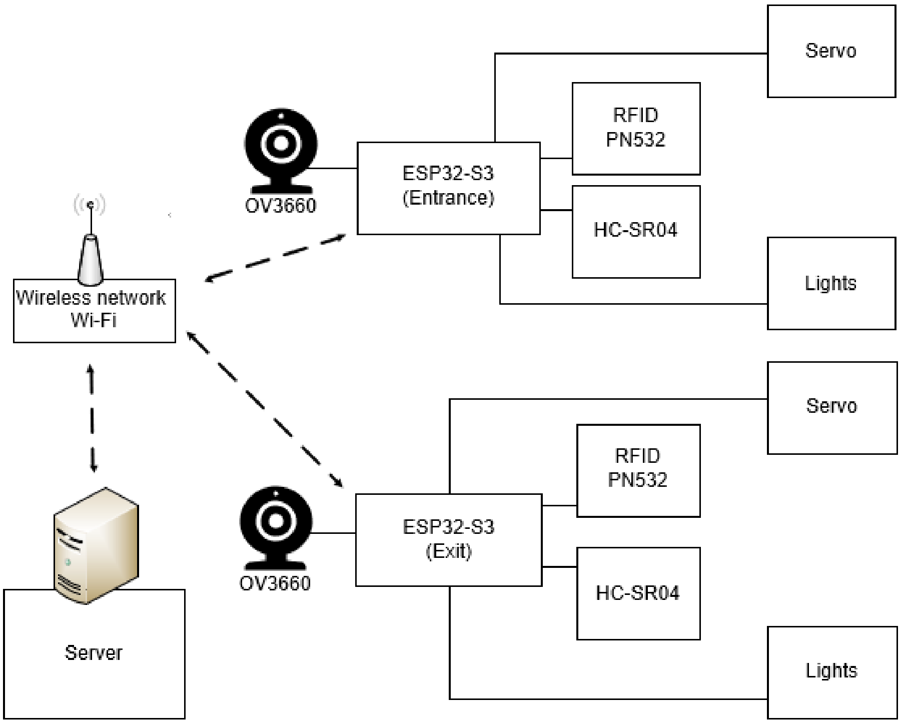


Fig. 2. System architecture, showing interactions between components.

Secondly, the firmware starts services: launch the embedded web server and REST endpoints used for device interaction and monitoring. Then enter the main loop that consists of capturing images and sending snapshots to the backend for recognition. Receiving access decisions (open/close) and actuating the barrier. Polling RFID events and forward them to backend routes for verification. Continuously log system activities and maintain health/status for diagnostics

The system operates as follows: when a vehicle approaches, the camera captures an image that the backend processes using a YOLOv8-based ALPR pipeline. The recognized plate is checked against an authorized (white) list in the database. If the plate is not authorized, the system supports manual authentication via the RFID module. A distance sensor enforces a safety interlock so the barrier only moves when it is safe to do so.

The backend is a critical component responsible for processing authentication requests and orchestrating access decisions. Implemented with Flask [13], it provides RESTful APIs for the microcontroller and the web interface, and uses YOLOv8 for ALPR, SQLite (via SQLAlchemy) for storage, and asynchronous threading for concurrent request handling **Fig. 3a.**

The backend receives image data from the ESP32 and pre-processes it (e.g., resizing, color conversions). Then runs the YOLOv8-based pipeline to detect vehicles and recognize license plates (detection + OCR in the region of interest) **Fig. 3b**. After matches the recognized plates against an authorized “whitelist” in the database. Logs all access attempts for security audits and exposes retrieval via API (e.g., /logs). Sends control commands (open/close) back to the ESP32 microcontroller over REST.

In addition to ALPR, the backend handles RFID-based events via dedicated endpoints, validates tags against the database, and logs outcomes for auditability. The use of RESTful APIs enables seamless communication with both the microcontroller and the web interface, while caching is employed to maintain responsiveness for real-time operation.

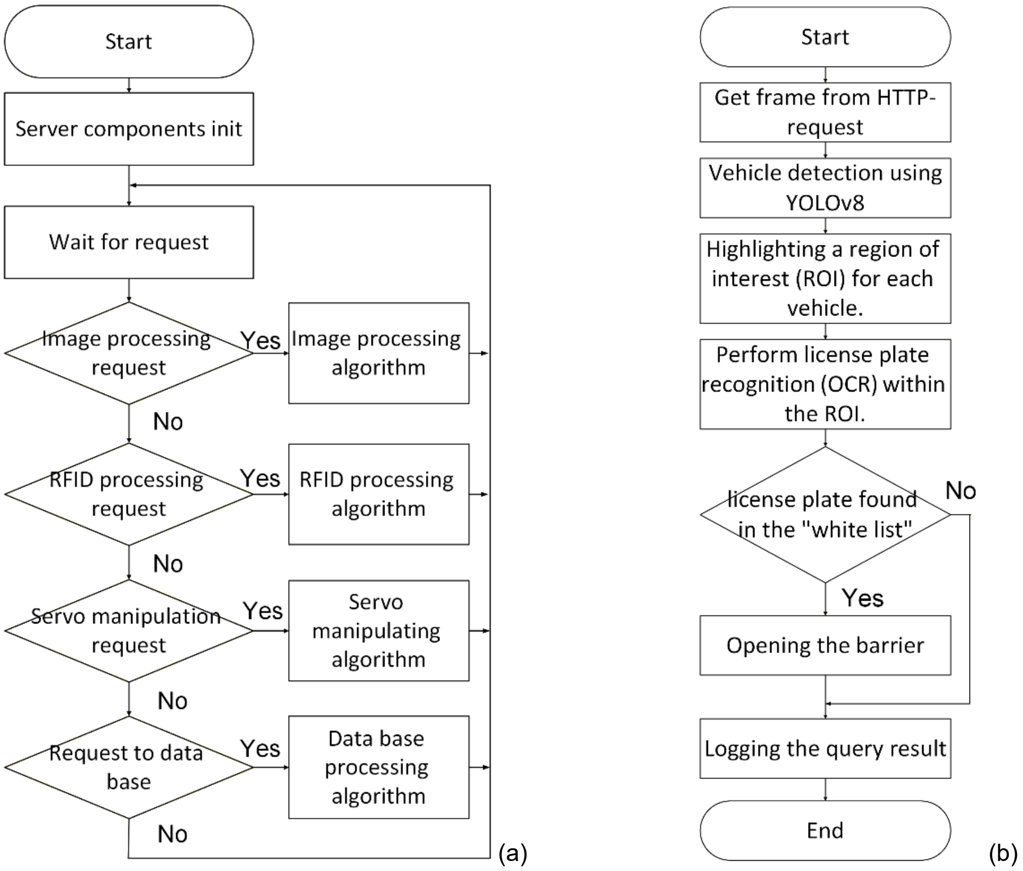
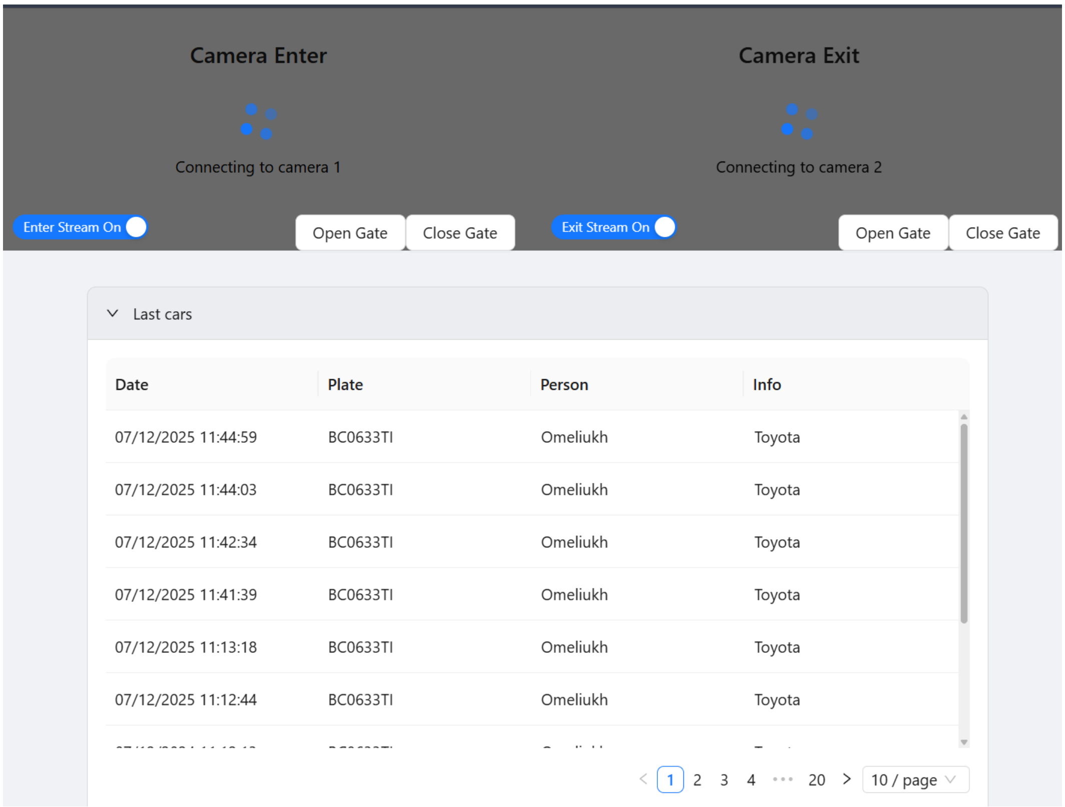

Start
Server components init
Wait for request
Image processing request
Yes
Image processing algorithm
No
RFID processing request
Yes
RFID processing algorithm
No
Servo manipulation request
Yes
Servo manipulating algorithm
No
Request to data base
Yes
Data base processing algorithm
No
(a)
Start
Get frame from HTTP-request
Vehicle detection using YOLOv8
Highlighting a region of interest (ROI) for each vehicle.
Perform license plate recognition (OCR) within the ROI.
license plate found in the "white list"
No
Yes
Opening the barrier
Logging the query result
End
(b)

Fig. 3. The backend scheme (a) and the image processing algorithm (b).

The system logs every access attempt (ALPR and RFID), stores audit trails, and provides retrieval via API for operator review in the web interface. Configuration parameters (e.g., safe distance) are synchronized via API to keep device and server settings consistent in real time.

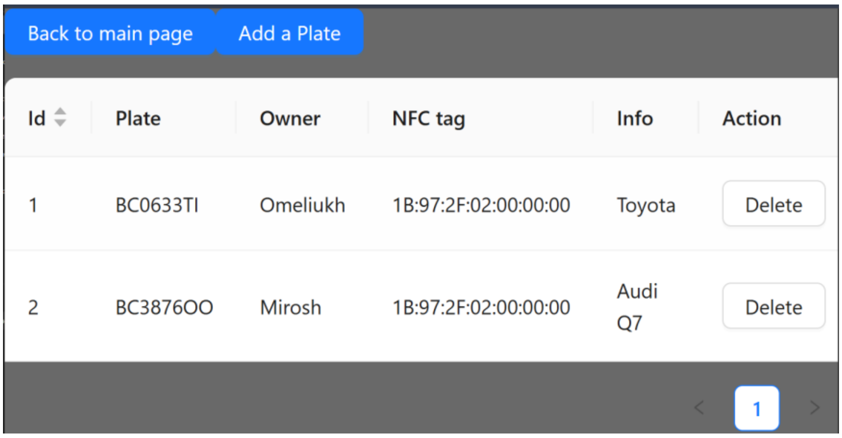
The web interface (React-based) enables real-time monitoring of camera feeds and logs, management of registered users/vehicles, system configuration (e.g., safe distance), and manual barrier control when required. The React-based web interface **Fig. 4** provides operators and administrators with a streamlined, real-time view of system activity and controls. It enables users to: view real-time access logs and monitor live camera feeds, with automatic reconnect attempts every five seconds to maintain continuity under network issues. Manage registered users, vehicles, and plate storage (e.g., “Cars and plates storage”), and review “Last cars” entries with timestamps, plate numbers, and associated info **Fig. 5**.

The interface implements authentication and session handling (e.g., isAuthenticated token and login redirect) and integrates with the Flask backend via RESTful APIs for log retrieval and device commands, supporting responsive real-time operation. Role-based access can be applied to ensure security personnel and administrators see only the features relevant to their responsibilities.



Date	Plate	Person	Info
07/12/2025 11:44:59	BC0633TI	Omeliukh	Toyota
07/12/2025 11:44:03	BC0633TI	Omeliukh	Toyota
07/12/2025 11:42:34	BC0633TI	Omeliukh	Toyota
07/12/2025 11:41:39	BC0633TI	Omeliukh	Toyota
07/12/2025 11:13:18	BC0633TI	Omeliukh	Toyota
07/12/2025 11:12:44	BC0633TI	Omeliukh	Toyota

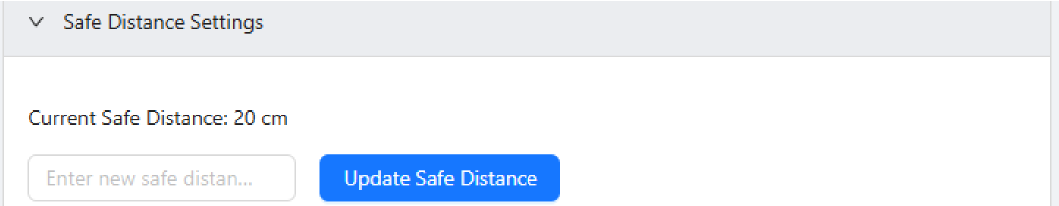
Fig. 4. Web interface screenshot displaying real-time vehicle monitoring and access control functions.



Id	Plate	Owner	NFC tag	Info	Action
1	BC0633TI	Omeliukh	1B:97:2F:02:00:00:00	Toyota	Delete
2	BC3876OO	Mirosh	1B:97:2F:02:00:00:00	Audi Q7	Delete

Fig. 5. Web interface “Cars and plates storage”.

Adjust system settings and security parameters, including safe-distance thresholds for barrier operation (e.g., “Current Safe Distance: 20 cm”). Manually override barrier control when necessary for operational continuity **Fig. 6**.



Safe Distance Settings

Current Safe Distance: 20 cm

Fig. 6. Web interface. Safe-distance thresholds.

RESULTS AND DISCUSSION

To validate the performance and reliability of the proposed system, a series of controlled experiments was conducted. The evaluation focused on two primary metrics: the accuracy of the Automatic License Plate Recognition (ALPR) module and the overall system response time.

To determine the accuracy of the YOLOv8-based ALPR pipeline, a custom dataset consisting of 1,000 static images was curated. This dataset was designed to reflect real-world operating conditions and included images captured at various times of day (daylight, dusk, and night), different weather conditions (clear, overcast), diverse viewing angles, and with partially obscured or soiled license plates.

The evaluation was performed using a Python script that processed each image in the dataset. A "successful" recognition was defined as an instance where the system correctly identified and transcribed all alphanumeric characters on the license plate in the correct order.

Out of the 1,000 test images, the system successfully recognized 750 plates, resulting in an overall accuracy of 75%. An analysis of the 250 failed cases revealed that the majority of errors occurred under specific challenging conditions: Low Light/Night: 42% of failures were due to insufficient illumination or heavy headlight glare. Obscured Plates: 30% of failures involved plates partially covered by dirt, snow, or tow hitches. Extreme Angles: 20% of failures occurred when the vehicle approached the camera at an angle greater than 40 degrees. Other: 8% of failures were due to motion blur or non-standard fonts on the plates.

System response time was defined as the end-to-end latency, encompassing image acquisition by the OV3660, Wi-Fi transmission – using 2.4 GHz 802.11n, backend processing – ALPR inference and database lookup, and the return transmission of the control command to the ESP32-S3. This metric was recorded under stable network conditions with a signal strength (RSSI) better than –65 dBm. System response time was recorded for all 1,000 test cases in the accuracy assessment. The resulting data was analyzed to determine the central tendency and variance. The arithmetic mean response time for the system was 820 ms. The standard error ±45 ms, indicating consistent performance with low latency across the test batch. This sub-second response time confirms the system's suitability for real-time operation without causing significant delays for users.

In addition to the primary metrics, functional tests were performed on auxiliary components. The PN532 module's reading range was confirmed to be reliable up to 5 cm at various card presentation angles (0, 45, and 90 degrees). The HC-SR04 ultrasonic sensor was validated by repeatedly placing obstacles at the barrier's path, confirming that it correctly halted barrier motion in 100% of test cases. All performance data was logged by the Flask backend application and analyzed using custom Python scripts utilizing the Pandas and NumPy libraries for statistical computation.

A critical aspect of system optimization was identifying the optimal camera resolution. This parameter presents a fundamental trade-off: higher resolutions (e.g., UXGA, 1600x1200) provide superior image detail for the ALPR model but drastically reduce the real-time processing speed, measured in average frames per second (FPS). Conversely, lower resolutions (e.g., QVGA, 320x240) are processed very quickly but often fail to capture enough detail for accurate recognition, significantly reducing the system's 75% accuracy.

A systematic test was conducted to measure the average FPS the system could achieve when processing streams at different standard resolutions. The results of this analysis are illustrated in **Fig. 7**. As the graph clearly shows, there is a sharp, non-linear drop in performance as resolution increases.

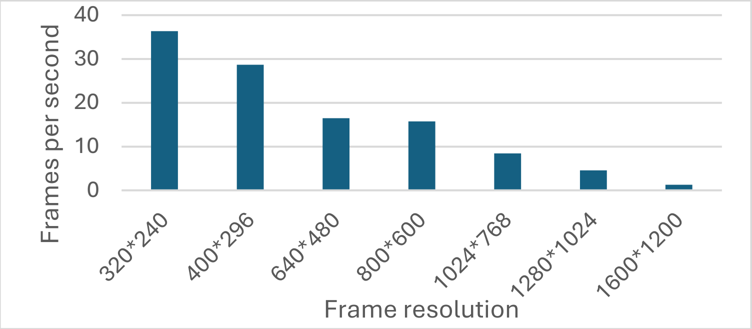


Fig. 7. Impact of image resolution on the system's average processing speed (FPS).

Based on this data, the 800x600 (SVGA) resolution was identified as the optimal trade-off. It provides a processing speed that is fast enough for responsive, real-time operation while preserving the image quality necessary for the ALPR model to function effectively.

CONCLUSION

This study successfully demonstrated the design, implementation, and practical validation of a cost-effective, automated access control system based on IoT principles. The key achievement of the work is the successful integration of a dual-modal authentication architecture, combining Automatic License Plate Recognition (ALPR) with a robust Radio-Frequency Identification (RFID) fallback, managed by a modern client-server backend.

The most significant results stem from the system's performance evaluation. The ALPR module, powered by a YOLOv8 model, achieved an accuracy of 75% on a challenging 1,000-image dataset. This quantitative result provides a realistic baseline for the system's effectiveness. Furthermore, the system demonstrated excellent real-time capabilities, with an average response time of 820 ms (±45 ms standard error), confirming its suitability for high-traffic environments where delays are unacceptable. The successful integration of an ultrasonic safety sensor, which reliably prevented barrier motion in 100% of test cases, underscores the system's readiness for practical deployment.

The practical significance of this research lies in its complete, end-to-end system design. By combining edge computing on the ESP32-S3 microcontroller with a flexible backend (Python Flask) and a real-time monitoring web interface (React), this work provides a scalable and affordable blueprint that significantly enhances security and efficiency compared to traditional manual access methods.

The analysis of ALPR failures (which led to 75% accuracy) clearly illuminates the path for future improvements. Failures were predominantly caused by adverse lighting conditions (headlight glare, low light) and significant plate obstructions. Therefore, future prospects for this research should focus on enhancing the robustness of the computer vision model. This can be achieved by re-training the model on a more diverse dataset featuring augmented night-time and poor-weather conditions, as well as exploring image pre-processing algorithms (e.g., local contrast enhancement) to improve image quality before recognition.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This work was supported by the Ministry of Education and Science of Ukraine (Project No. 0125U001883).

COMPLIANCE WITH ETHICAL STANDARDS

The authors declare that they have no competing interests.

AUTHOR CONTRIBUTIONS

Conceptualization, [N.O., H.K., T.T]; methodology, [N.O.]; investigation, [N.O.]; writing – original draft preparation, [N.O., H.K.]; writing – review and editing, [N.O., H.K.]; visualization, [N.O.].

All authors have read and agreed to the published version of the manuscript.

REFERENCES

[1] Bamashmos, S., Chilamkurti, N., & Shahraki, A. S. (2024). Two-Layered Multi-Factor Authentication Using Decentralized Blockchain in an IoT Environment. Sensors, 24(11), 3575. https://doi.org/10.3390/s24113575

[2] Babii, A., & Samila, A. (2023). Dual Authentication Technique for RFID Access Control Systems with Increased Level of Protection. Security of Infocommunication Systems and Internet of Things, 1(1), Article 01011. https://doi.org/10.31861/sisiot2023.1.01011

[3] Riad, K. (2025). Robust and Leakage-Resilient Access Control for IoT Outsourcing with Attribute-Based Encryption. Sensors, 25(3), 625. https://doi.org/10.3390/s25030625

[4] Albugmi, A. (2025). Hybrid smart IoT detection and prevention framework for smart cities using blockchain technology. International Journal of Advanced and Applied Sciences, 12(4), 107-115. https://doi.org/10.21833/ijaas.2025.04.013

[5] Wang, X., Wang, M., Guo, H., Li, J., Wang, X., & Zhang, Y. (2025). License plate recognition system for complex scenarios based on improved YOLOv5s and LPRNet. Scientific Reports, 15, Article number 34741. https://doi.org/10.1038/s41598-025-18311-4

[6] Fadlianda, D., Fikry, M., & Nunsina. (2024). Innovative IoT-Based Automatic Gate System with RFID and Electro-Magnetic Lock for Secure Access. Proceedings of Malikussaleh International Conference on Multidisciplinary Studies (MICoMS), 4, Article 884. https://doi.org/10.29103/micoms.v4i.884

[7] Pradhan, G., Prusty, M. R., Negi, V. S., Chinara, S., et al. (2025). Advanced IoT-integrated parking systems with automated license plate recognition and payment management. Scientific Reports, 15, Article number 2388. https://doi.org/10.1038/s41598-025-86441-w

[8] Kokila, M., & Srinivasa Reddy, K. (2024). Authentication, access control and scalability models in Internet of Things security – A review. Cyber Security and Applications, 10, 123-146. https://doi.org/10.1016/j.csa.2024.100057

[9] Kalamaras, S. D., Tsitsimpikou, M.-A., Tzenos, C. A., Lithourgidis, A. A., Pitsikoglou, D. S., & Kotsopoulos, T. A. (2025). A Low-Cost IoT System Based on the ESP32 Microcontroller for Efficient Monitoring of a Pilot Anaerobic Biogas Reactor. Applied Sciences, 15(1), 34. https://doi.org/10.3390/app15010034

[10] Chang, Y.-H., Wu, F.-C., & Lin, H.-W. (2025). Design and Implementation of ESP32-Based Edge Computing for Object Detection. Sensors, 25(6), 1656. https://doi.org/10.3390/s25061656

[11] Babii, A., & Samila, A. (2023). Dual Authentication Technique for RFID Access Control Systems with Increased Level of Protection. SISIOT, 1(1), 01011. https://doi.org/10.31861/sisiot2023.1.01011

[12] Syahputra, D., Adriansyah, A., & Wibowo, A. T. (2023). Performance sensor analysis of HC-SR04 proximity sensor on distance measuring device with fuzzy logic method. JEEMECS: Journal of Electrical Engineering, Mechatronic and Computer Science, 7(1), 1-7. https://doi.org/10.26905/jeemecs.v7i1.10096

[13] Sabbatini, M. (2024). Hardening IoT Devices: An Analysis of the ESP32-S3 Microcontroller. (Master's thesis). University of Zurich. https://doi.org/10.1109/ACCESS.2021.3092938

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ ДОСТУПУ НА БАЗІ ІНТЕРНЕТУ РЕЧЕЙ З ІНТЕГРАЦІЄЮ ТЕХНОЛОГІЙ АВТОМАТИЧНОГО РОЗПІЗНАВАННЯ НОМЕРНИХ ЗНАКІВ ТА РАДІОЧАСТОТНОЇ ІДЕНТИФІКАЦІЇ

Назар Омелюх , ***Галина Клим**** , ***Тарас Ткачук***
Національний університет «Львівська політехніка»,
вул. Бандери 12, 79013 м. Львів, Україна

АНОТАЦІЯ

Обґрунтування. Забезпечення безпеки приватних територій стає дедалі актуальнішим завданням у сучасному світі. Традиційні методи контролю, що покладаються на фізичну охорону або механічні бар'єри з використанням ключів, є малоефективними та вразливими до ризиків, таких як крадіжка або дублювання перепусток. Крім того, застарілі системи не забезпечують належного аудиту подій. Це формує нагальну потребу у сучасних автоматизованих IoT-системах, які гарантують надійне управління доступом та моніторинг у реальному часі.

Матеріали та методи. Ядром системи є недорогий мікроконтролер з модулем камери. Зчитувач радіочастотної ідентифікації (RFID) сканує картки доступу. Ультразвуковий сенсор відстані виявляє перешкоди для безпеки, а сервопривід керує фізичним шлагбаумом. Серверне програмне забезпечення (бек-енд) було розроблено на Python, а веб-панель керування – на JavaScript (React).

Система поєднує два методи ідентифікації. По-перше, камера фіксує зображення транспортного засобу, надсилаючи його на сервер, де модель ШІ зчитує номерний знак. Сервер перевіряє номер у списку дозволених. Якщо номер не розпізнано, водій сканує RFID-картку як вторинний метод. Сенсор відстані постійно контролює зону шлагбаума, щоб запобігти закриттю на перешкоді. Веб-інтерфейс дозволяє оператору спостерігати за камерою, переглядати журнали та вручну управляти шлагбаумом.

Результати. Розроблена система була успішно протестована. Модель ШІ досягла 75% точності в ідентифікації номерних знаків. Система виявилася швидкою, із середнім часом відгуку від зйомки до прийняття рішення менше однієї секунди. Датчик безпеки був перевірений, надійно виявляючи перешкоди та запобігаючи руху шлагбаума, що забезпечує безпечну роботу. Були представленні результати порівняння якості відео та часом відгуку системи. Оптимальний баланс між якістю відео та часом відгуку було знайдено при роздільній здатності 800x600.

Висновки. Було успішно розроблено, побудовано та протестовано надійну, економічно ефективну автоматизовану систему контролю доступу. Поєднання зчитування номерних знаків на основі ШІ з резервною системою RFID забезпечує надійне та гнучке рішення. Ця система добре підходить для покращення безпеки та ефективності в реальних умовах, таких як житлові, офісні та промислові зони.

Ключові слова: автоматизований контроль доступу, ESP32-S3, ALPR, RFID, безпека.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
28 November, 2025	15 December, 2026	16 December, 2025	25 December, 2025