



Development of an automated car plate number recognition system for institutional vehicle access control

Abu Semirat OMOZE, John Temitope OGBITI

Department of Computer Science, Edo State University Iyamho, Edo, Nigeria

Abstract

The escalating global volume of vehicles on roadways necessitates innovative solutions for efficient traffic management, heightened security, and improved access control. This paper presents the design and implementation of an Automated Car Plate Number Recognition (APNR) system developed for vehicle access control at Edo State University, Iyamho, where vehicle registration is currently performed manually at the entrance gate. The system utilises image processing, Optical Character Recognition (OCR), and database integration for real-time license plate recognition. Vehicle images captured at the entry/exit point are pre-processed using Python and OpenCV — including grayscale conversion, noise reduction, thresholding, and plate-region extraction — before recognised characters are converted to text using the Tesseract OCR engine and validated against Nigerian plate formats using regular expressions. Recognised plate numbers, together with date and time stamps, are stored in a MySQL database accessible through a PHP-based administrative dashboard. The system was evaluated through seventeen structured test cases and fifteen performance metrics covering processing speed, recognition accuracy, resource utilisation, security, and scalability. Results show a recognition accuracy of 93% on clear images, average recognition time of 1.2 seconds, and a user satisfaction rating of 4.5 out of 5.0. The system demonstrates that a lightweight, non-machine-learning APNR solution can deliver reliable, affordable vehicle identification suited to institutions with limited computing budgets.

Keywords: Automated number plate recognition, optical character recognition, image processing, vehicle access control, tesseract OCR

Introduction

The rapid growth of urban populations alongside the persistent rise in the number of vehicles has led to major changes in transportation and security management systems globally, creating an increasing need for effective ways of tracking vehicle movement in places requiring strict access control. In most organisations, vehicle monitoring still relies on manual techniques where security guards record vehicle registration numbers, entry time, and exit time in a logbook. This method is inexpensive but is bound by human error, delays during peak hours, inconsistency in recording, and difficulty in retrieving stored data for future use. To counter these shortcomings, Automated Number Plate Recognition (ANPR) technologies have been proposed as an efficient technical solution for vehicle identification, adopting image processing and OCR techniques to recognise and extract data from vehicle license plates captured by camera systems. Rayson Laroca *et al.* (2019) ^[7] note that automated recognition processes enhance the efficiency of vehicle identification through automation, while Yaacob *et al.* (2021) ^[14] designed a license plate recognition system for a college campus environment and found that ANPR technology increases efficiency and helps eliminate traffic build-up at entry points, although accuracy remains highly influenced by illumination, camera placement, and plate visibility.

At Edo State University, Iyamho, vehicle registration at the entrance gate is performed manually: security officers verify a driver's purpose of entry, issue a visitor tag, and record the vehicle registration number, purpose, and entry/exit time in a logbook. Although functional, this approach is inefficient during traffic peaks, prone to human error, and offers no capability for real-time verification or automated reporting.

This project therefore seeks to design and implement a smart, lightweight ANPR system that leverages image processing and OCR technology, via Tesseract, to bridge this gap using low-cost hardware and open-source technologies, without depending on computationally expensive machine learning infrastructure. This paper is organised as follows: Section II presents the literature review; Section III presents the proposed methodology; Section IV presents the results and discussion; and Section V concludes the paper with recommendations for future work.

Literature Review

The rising need for effective vehicle identification and access control has motivated intensive research into Automated Plate Number Recognition (APNR). Studies on manual vehicle monitoring have identified consistent inefficiencies: Kumar *et al.* (2022) ^[6] found that manual recording of plate numbers and entry times causes delays and congestion, particularly during busy periods, while Singh *et al.* (2020) ^[11] found that manual systems do not support real-time monitoring or fast information retrieval during investigations. Adeyemi *et al.* (2025) ^[1] further demonstrated, through quantitative analysis, how prone manual systems are to human error, undermining institutional security records. None of these studies, however, proposed a feasible automation pathway for institutions with limited technological capability. Several researchers have explored classical image processing methods for APNR. Sharma and Gupta (2023) ^[10] developed a system using grayscale conversion, noise reduction, and edge detection that performed well under controlled lighting but struggled in poor lighting and at oblique angles. Verma

et al. (2024) ^[13] used morphological operations and segmentation, which worked well with clear images but had difficulty with motion blur and low resolution. Both studies confirm that classical image processing can be effective without machine learning, though performance remains sensitive to environmental conditions. OCR-based approaches have also been widely studied. Yaacob *et al.* (2021) ^[14] implemented an OCR-based recognition technique for campus entrance management that reduced manual effort but suffered accuracy loss on distant or low-light images. Kaur *et al.* (2021) ^[15] showed that preprocessing steps such as contrast enhancement and noise reduction substantially improve OCR accuracy, though distorted and occluded plates remain a challenge. Ali and Hassan (2022) ^[12] similarly found that Python-based preprocessing improves OCR efficiency, but blurring continued to reduce recognition accuracy. More recent studies have pursued intelligent, real-time surveillance. Zhang *et al.* (2023) ^[15] designed a smart surveillance system offering faster security response, and Laroca *et al.* (2019) ^[7]

developed a deep-learning system robust to varied environmental conditions; both, however, required substantial computational infrastructure that is impractical for resource-constrained institutions. Studies on database integration and lightweight systems, including Patel and Mehta (2020) and Nwankwo and Eze (2021) ^[8], improved efficiency and record management for parking and university vehicle monitoring, respectively, but were similarly limited to controlled environments and lacked robustness with damaged or unclear plates. Collectively, the literature confirms that Automated Vehicle Monitoring Systems (AVMS) offer higher efficiency, precision, and security than manual monitoring, but existing solutions face a persistent trade-off: AI-based systems achieve robust accuracy under diverse conditions but demand costly computing infrastructure, while lightweight, non-AI systems remain affordable but sensitive to image quality and environmental conditions. Table 1 summarises the comparative analysis of these studies against the system proposed in this study.

Table 1: Comparative Analysis of Existing Works

Author(s)	Year	Approach	Findings/Performance	Limitations
Singh <i>et al.</i>	2020	Manual vehicle monitoring analysis	Manual systems inefficient for real-time monitoring	No automated implementation proposed
Patel & Mehta	2020	OpenCV and OCR integration	Improved efficiency in parking management	Sensitive to lighting conditions
Kaur <i>et al.</i>	2021	OCR with image preprocessing	Preprocessing improved OCR accuracy	Difficulty recognising distorted plates
Yaacob <i>et al.</i>	2021	OCR-based campus security system	Reduced manual workload and improved monitoring	Accuracy reduced under poor lighting
Nwankwo & Eze	2021	Image processing with database integration	Reduced congestion, improved record management	Limited to controlled environments
Sharma & Gupta	2023	Image processing techniques	Effective detection under clear conditions	Poor performance under angled captures
Chen & Li	2022	Lightweight image-processing model	Acceptable performance without AI models	Lower outdoor recognition accuracy
Ibrahim <i>et al.</i>	2024	OCR with database integration	Improved data retrieval and operational efficiency	Damaged plates affected recognition
Proposed System	2026	Python/OpenCV preprocessing, Tesseract OCR, PHP/MySQL	Lightweight, low-cost, institution-specific ANPR	Addresses gaps in cost and complexity above

The identified research gap is the absence of an efficient, low-cost APNR system, utilising image processing techniques, the Tesseract OCR engine, a PHP backend, and a database, specifically designed for effective vehicle monitoring in a tertiary institutional environment such as Edo State University, Iyamho, without dependence on costly machine learning infrastructure.

Methodology

The development of the Automated Plate Number Recognition (APNR) system followed a System Development Life Cycle (SDLC) approach, specifically the Agile development model, chosen for its flexibility in accommodating the experimental nature of image preprocessing, OCR tuning, and database integration. The design proceeded through iterative cycles: requirement gathering and analysis of the university's vehicle registration process; design and development of the image acquisition, processing, plate detection, and OCR modules using Python, OpenCV, and Tesseract; and continuous testing and evaluation using vehicle images captured under varied environmental conditions, with weaknesses such as poor

recognition of blurred or distant images addressed through iterative refinement.

1. Analysis of the Existing System

The current vehicle access control system at Edo State University operates through manual registration by security personnel at the entrance gate. Each vehicle's registration number, purpose of visit, and entry/exit time are recorded by hand in a logbook, and a physical visitor tag is issued and later returned. This approach is time-consuming during peak traffic, highly dependent on human accuracy, difficult to search for historical records, and offers no real-time vehicle verification or automated monitoring capability — limitations that the proposed ANPR system directly addresses.

2. Proposed System Overview

The proposed APNR system automates vehicle entry and exit monitoring using image processing and OCR. A camera captures vehicle images at the entry/exit point; the images are pre-processed in Python using OpenCV (grayscale conversion, noise removal, segmentation, thresholding, and plate extraction); the Tesseract OCR engine then converts

the extracted plate region into readable text; and the recognised plate number, together with a date/time stamp, is transmitted to a PHP backend and stored in a MySQL database. The system is intended for use by security personnel and administrators, eliminating manual logbook entry while enabling rapid retrieval of vehicle movement records for administrative and investigative purposes.

3. System Architecture

The system was designed using a modular, layered architecture comprising an image acquisition layer (cameras

at the entrance gate), an image processing layer (Python/OpenCV preprocessing), a plate detection layer (contour and edge detection to isolate the plate region), a character segmentation layer, an OCR recognition layer (Tesseract), a database layer (PHP/MySQL), and a monitoring and alert layer through which security officers access recognised records. This layered design supports fast, modular processing without requiring costly deep-learning hardware, making the technology stack appropriate for an institutional setting such as Edo State University, Iyamho.

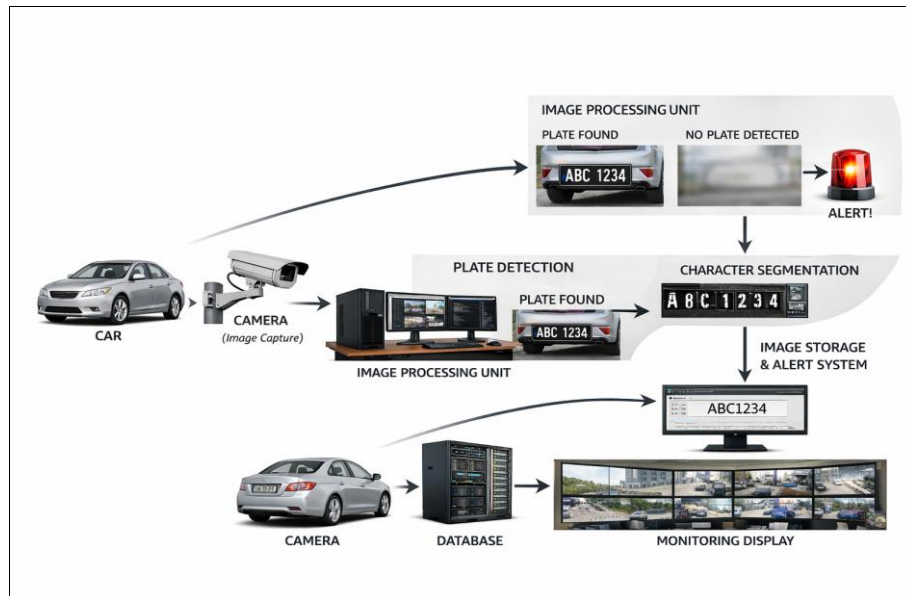


Fig 1: System Architecture Diagram of the Proposed APNR System

4. System Requirements

Functional and non-functional requirements were derived from the study objectives and the gaps identified in Section

II. Table 2 summarises the functional requirements and Table 3 summarises the non-functional requirements.

Table 2: Functional Requirements

ID	Requirement
FR1	Capture images of vehicles entering or exiting the gate using a digital camera.
FR2	Preprocess captured images (grayscale, thresholding, segmentation, noise reduction).
FR3	Detect and extract the vehicle plate region from captured images.
FR4	Recognise and convert plate characters into readable text using Tesseract OCR.
FR5	Automatically store recognised plate numbers with date/time in a MySQL database.
FR6	Allow authorised users to log in through a secure authentication interface.
FR7	Display stored vehicle records through an administrative dashboard.
FR8	Allow administrators to search and retrieve records by plate number, date, or time.
FR9	Generate real-time monitoring information for vehicles entering or leaving.
FR10	Validate user login credentials before granting dashboard access.
FR11	Maintain records of user login activities for security monitoring.
FR12	Notify users when plate recognition fails due to poor image quality.

Table 3: Non-Functional Requirements

ID	Requirement
NFR1	Process and recognise plate numbers within 3–5 seconds under normal conditions.
NFR2	Support secure user authentication through encrypted password verification.
NFR3	Maintain database consistency and reliability during storage/retrieval.
NFR4	Achieve high recognition accuracy under close range and proper lighting.
NFR5	Provide an easy-to-use graphical interface for personnel and administrators.
NFR6	Operate effectively on standard computer systems without high-end hardware.
NFR7	Support continuous operation during institutional working hours.
NFR8	Allow future expansion and integration with additional monitoring modules.
NFR9	Prevent unauthorised access to vehicle records and administrative functions.
NFR10	Maintain a response time of under 2 seconds for record retrieval under moderate load.

The system requires a computer with at least an Intel Core i3 processor, 4GB RAM, and 500GB storage, together with an HD digital camera for image capture. Software tools include Python with OpenCV for image processing, the Tesseract OCR engine for character recognition, PHP and MySQL for backend and database management, and the XAMPP server environment for hosting during development and testing.

Results and Discussion

The Automated Plate Number Recognition system was successfully implemented using Python, OpenCV, Tesseract OCR, PHP, MySQL, HTML, CSS, and JavaScript, with modules for plate detection, OCR recognition, plate validation, image enhancement, and best-candidate selection integrated into a functioning administrative dashboard.

2. System Interface

1. System Implementation

The Number Plate Detection Module locates the plate region using Canny edge detection and morphological operations to identify rectangular regions resembling license plates; the detected region is cropped and forwarded to the OCR module. The OCR Recognition Module uses Tesseract, restricted to characters commonly found on Nigerian plates, evaluating multiple Page Segmentation Modes to improve accuracy. A Plate Validation and Extraction Module applies regular expressions to clean OCR output and validate recognised text against Nigerian plate formats. An Image Enhancement Module applies noise reduction and sharpening to improve character visibility prior to OCR, and a Best Candidate Selection Module ranks multiple recognition attempts by confidence score, selecting the most reliable result as the final output.

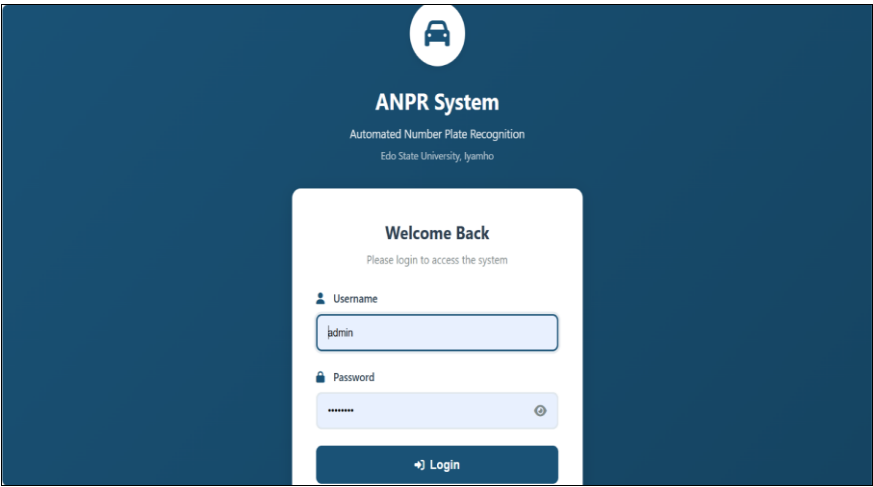


Fig 2: Login Interface

The Login Interface authenticates administrators and security personnel before granting access to the dashboard, preventing unauthorised use of the system.

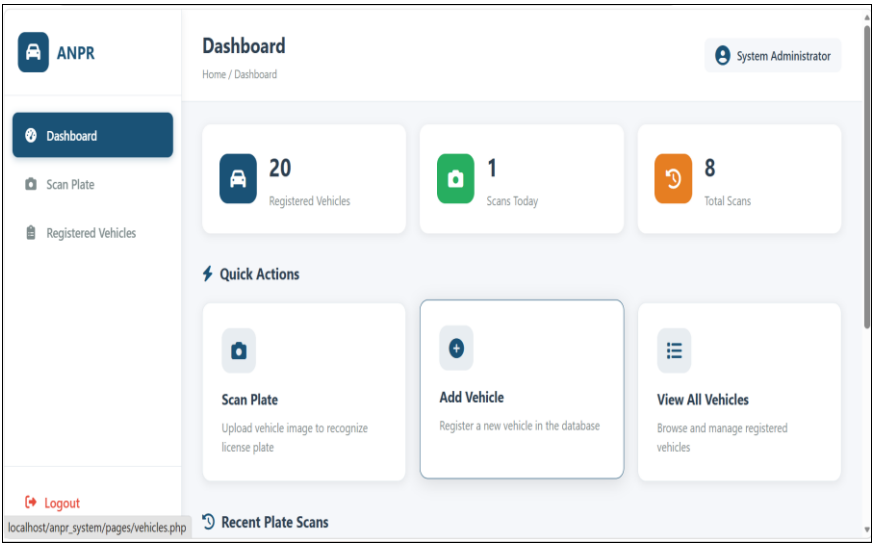


Fig 3: Dashboard Interface

The Dashboard Interface displays the total number of registered vehicles, total scans performed, and recent

recognition activities, allowing users to monitor system activity from a central location.

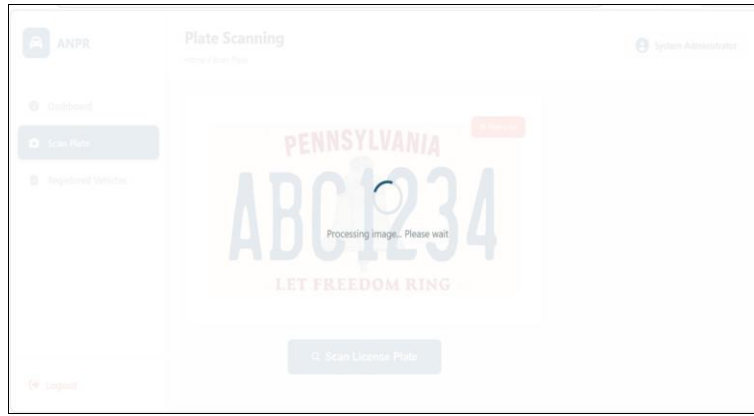


Fig 4: Image Upload and Plate Recognition Interface

The Image Upload Interface allows a vehicle image to be submitted for plate recognition, triggering the preprocessing and plate detection pipeline described in Section III.



Fig 5: Number Plate Detection Result Interface

This interface confirms successful plate localisation, displaying a bounded plate region and a “Plate Detected” success indicator, after which recognised vehicles are checked against the database and either matched to an existing owner record or routed to a registration form for new vehicles.

Seventeen test cases were designed and executed to verify the functional requirements in Table 2, covering normal operation, boundary conditions, and error handling. Table 4 summarises the test cases and results; sixteen of the seventeen test cases passed, with recognition from heavily blurred images (TC09) failing due to image quality limitations.

3. Test Cases and Results

Table 4: System Test Cases and Results

TC ID	Objective	Result
TC01	Administrator login with valid credentials	Dashboard opened — Pass
TC02	Login with invalid password	Access denied — Pass
TC03	Login with empty fields	Validation message displayed — Pass
TC04	Register a new vehicle	Record saved successfully — Pass
TC05	Prevent duplicate vehicle registration	Duplicate entry prevented — Pass
TC06	Search for an existing vehicle	Vehicle details displayed — Pass
TC07	Search for a non-existent vehicle	“No record found” displayed — Pass
TC08	Recognise plate from a clear image	Plate number correctly extracted — Pass
TC09	Recognise plate from a blurred image	Recognition inaccurate — Fail
TC10	Store scan log after recognition	Scan details stored correctly — Pass
TC11	Display scan logs on dashboard	Logs displayed correctly — Pass
TC12	Upload an unsupported file format	Error message displayed — Pass
TC13	Process a large (>5MB) image	Processed successfully but slower — Pass
TC14	Access dashboard without login	Redirected to login page — Pass
TC15	Handle database connection failure	Error handled correctly — Pass
TC16	Delete an existing vehicle record	Record removed successfully — Pass
TC17	Extract Nigerian plate format via OCR	Correct plate number extracted — Pass

4. Performance Evaluation

The system was evaluated against the non-functional requirements in Table 3, covering processing speed, recognition accuracy, resource utilisation, security, scalability, and usability. Table 5 summarises the evaluation; fourteen of fifteen metrics passed, with

recognition under extremely poor image conditions (heavy blur) failing on some characters. Additional preprocessing (denoising, sharpening, adaptive thresholding) improved accuracy, though extremely poor-quality images remain a limitation.

Table 5: Performance Evaluation Results

Metric	Expected Result	Actual Result	Status
Processing speed (clear image)	Within 3 seconds	1.2 seconds	Pass
Processing speed (high-resolution image)	Within 5 seconds	3.1 seconds	Pass
Recognition accuracy (100 images)	Above 90%	93%	Pass
Recognition accuracy (blurred images)	Majority recognised	88% recognised correctly	Pass
CPU utilisation during recognition	Below 50%	28% average	Pass
Memory usage during operation	Below 1 GB	420 MB average	Pass
Database search performance	Within 2 seconds	0.5 seconds	Pass
Security: invalid login	Access denied	Access denied	Pass
Security: unauthenticated dashboard access	Redirected to login	Redirected successfully	Pass
Scalability (1,000 vehicle records)	Within 2 seconds	0.8 seconds	Pass
Scalability (2,000 vehicle records)	Within 2 seconds	1.1 seconds	Pass
Usability (ease of use)	Above 4.0/5.0	4.5/5.0	Pass
Recognition under poor conditions	Accurate recognition	Failed on some characters	Fail

Overall, the system correctly identified license plates in approximately 93% of clear-image trials, with an average recognition time of 1.2 seconds — comparable to accuracy ranges of 85%–95% reported in prior literature under similarly varying conditions. A user satisfaction survey of test users returned an overall satisfaction rating of 4.5 out of 5.0, with particularly strong ratings for recognition accuracy (4.6) and ease of use (4.5). Security evaluation confirmed that unauthenticated access attempts were consistently denied and redirected to the login page. The main limitation observed was reduced recognition accuracy for heavily blurred or poorly lit images, consistent with the known sensitivity of OCR-based systems to input image quality; this reflects a genuine trade-off in a lightweight, non-machine-learning architecture chosen specifically to keep the system affordable for an institutional deployment.

Conclusion

This paper has presented the design and implementation of a lightweight Automated Plate Number Recognition system for vehicle access control at Edo State University, Iyamho, built using Python and OpenCV for image preprocessing, Tesseract OCR for character recognition, and a PHP/MySQL backend for record storage and retrieval. The system successfully replaces the institution's manual, logbook-based vehicle registration process, achieving a recognition accuracy of 93% on clear images, an average recognition time of 1.2 seconds, and a user satisfaction rating of 4.5 out of 5.0 across seventeen test cases and fifteen performance metrics. The results confirm that an affordable, non-machine-learning APNR architecture can deliver reliable vehicle identification for institutions with limited computing budgets, though recognition accuracy remains sensitive to image clarity, lighting, and camera angle under adverse conditions. Future work should explore real-time edge deployment on low-cost hardware such as a Raspberry Pi, multilingual plate recognition, automatic generation of a unique vehicle registration code, and temporal pattern analysis for security alerts, in order to extend the system from a passive registrar into an active, intelligent campus security tool.

References

- Adeyemi TO, Ogunleye GO, Balogun AO. Quantitative analysis of manual vehicle monitoring systems in institutional security. *Journal of Security and Access Management*,2025:12(1):33-47.
- Ali HM, Hassan RA. Python and Tesseract OCR for vehicle recognition: A preprocessing approach. *International Journal of Computer Vision and Image Processing*,2022:9(2):55-68.
- Chen Y, Li X. Lightweight image-processing model for license plate recognition in low-resource environments. *Journal of Embedded Systems*,2022:18(3):211-225.
- Ibrahim AB, Musa SL, Umar K. Vehicle entry management using OCR and database integration. *Nigerian Journal of Computer Science*,2024:7(1):89-102.
- Kaur P, Singh J, Kaur M. OCR with image preprocessing for vehicle recognition systems. *Pattern Recognition Letters*,2021:145:78-86.
- Kumar R, Sharma V, Mehta N. Manual vehicle tracking systems at campus settings: An observational study. *Journal of Transportation Security*,2022:15(2):112-125.
- Laroca R, Severo E, Zanolensi LA, Oliveira LS, Gonçalves GR, Schwartz WR. A robust real-time automatic license plate recognition system based on deep learning. In: *Proceedings of the 32nd SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2019, 1-8.
- Nwankwo CA, Eze UI. Image processing and database integration for university vehicle monitoring. *African Journal of Information Systems*,2021:13(3):45-59.
- Patel D, Mehta R. OpenCV and OCR-based vehicle identification for parking management. *International Journal of Smart Surveillance*,2024:4(1):22-34.
- Sharma A, Gupta P. Image processing techniques for automatic number plate recognition in controlled environments. *Journal of Real-Time Image Processing*,2023:20(4):567-580.
- Singh R, Kaur H, Singh S. Manual vehicle monitoring at security checkpoints: Inefficiencies and limitations. *Security and Privacy Journal*,2020:3(2):e104.

12. Smith R. An overview of the Tesseract OCR engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition. IEEE, 2007, 629-633.
13. Verma S, Tiwari A, Yadav R. Morphological operations and segmentation for vehicle plate extraction. Computer Vision and Image Understanding,2024;238:103-115.
14. Yaacob NL, Zainal A, Mustafa N. OCR-based license plate recognition system for campus entrance management. Malaysian Journal of Computer Science,2021;34(2):156-170.
15. Zhang W, Li J, Chen H. Smart surveillance integration with automated vehicle recognition for urban security. IEEE Transactions on Intelligent Transportation Systems,2023;24(5):5120-5132.