

An Integrated YOLO–OCR Architecture for Real-Time Automatic Number Plate Recognition (ANPR) in Intelligent Access Management

Hisham Aboulghasem Ali Esherwi *

Department of Automatic Control, College of Computer Technology-Tripoli, Tripoli, Libya

معمارية متكاملة بين YOLO و OCR للتعرف الآلي على لوحات المركبات في الزمن الحقيقي ضمن أنظمة إدارة الوصول الذكية

هشام أبو القاسم علي الشروي *

قسم التحكم الآلي، كلية تقنية الحاسوب طرابلس، طرابلس، ليبيا

*Corresponding author: hisham.esherwi@gmail.com

Received: January 14, 2026

Accepted: February 25, 2026

Published: March 08, 2026



Copyright: © 2026 by the authors. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract:

This project presents a real-time Automated Number Plate Recognition (ANPR) system developed to support intelligent vehicle access management in controlled and security-sensitive environments. The proposed solution combines image processing with instant decision-making to enable a secure, contactless, and autonomous gate control mechanism. It responds to the increasing demand for efficient access systems, particularly in high-security locations such as institutional facilities and residential compounds. The framework is designed to be scalable and practical for deployment in real-world scenarios through a tight integration of hardware and software components. The detection module relies on YOLOv11, which was fine-tuned to accurately localize license plates under varying conditions. For character recognition, EasyOCR was adopted to reliably read both Arabic and English plate formats. A PostgreSQL database serves as the backend layer, enabling structured storage of plate records and efficient log management. Development was carried out using Visual Studio Code for Python scripting, debugging, and project organization. The overall architecture integrates ESP32-CAM units for image acquisition, Python-based real-time processing, and a user-friendly dashboard for monitoring captured plates and reviewing logs.

To validate performance, the system was evaluated across diverse operational settings, including low-light environments and high-glare scenes, to ensure robustness and stability. The results demonstrated fast response times and low latency in both detection and recognition tasks, alongside smooth communication between the user interface and the database layer. Overall, the final implementation delivers a secure, efficient, and cost-effective access control solution that can be adapted for a wide range of public and private surveillance and entry-management applications.

Keywords: Automatic Number Plate Recognition (ANPR), YOLO, Optical Character Recognition (OCR), Intelligent Access Control, Real-Time Systems.

المخلص

يعرض هذا المشروع نظامًا متقدمًا للتعرف الآلي على لوحات المركبات (ANPR) يعمل في الزمن الحقيقي، ويستهدف إدارة دخول المركبات بذكاء داخل البيئات المقيدة ذات الحساسية الأمنية. تقوم الفكرة الرئيسية على دمج تقنيات معالجة الصور مع آليات اتخاذ القرار الفوري، بما يتيح إنشاء حل آمن وغير تلامسي وذاتي التشغيل للتحكم في بوابات الدخول. ويأتي هذا

التوجه استجابةً للحاجة المتزايدة إلى أنظمة دخول فعالة، خاصة في المواقع التي تتطلب مستوى مرتفعاً من الحماية مثل مداخل المؤسسات والمجمعات السكنية. كما صُمم النظام ضمن إطار قابل للتوسع وسهل التطبيق ميدانياً عبر التكامل بين العناصر العتادية والبرمجية.

يعتمد النظام في مرحلة الكشف عن اللوحات على نموذج YOLOv11 بعد ضبطه وتحسينه لرفع دقة تحديد مواقع اللوحات في ظروف تشغيل مختلفة. وفي مرحلة قراءة الأحرف، تم توظيف مكتبة EasyOCR للتعرف على اللوحات العربية والإنجليزية بدرجة عالية من الاعتمادية. كما استُخدمت قاعدة بيانات PostgreSQL كطبقة خلفية لتخزين بيانات اللوحات وتنظيم السجلات وإدارتها بكفاءة. وقد تم تنفيذ التطوير باستخدام Visual Studio Code لدعم برمجة Python وتتبع الإصدارات وتصحيح الأخطاء. وتتكون معمارية النظام من وحدات ESP32-CAM لالتقاط الصور، ومعالجة آنية للصور باستخدام Python، إضافة إلى لوحة تحكم تفاعلية تُمكن المستخدم من استعراض البيانات والمتابعة ومراجعة السجلات. تم اختبار النظام في بيئات تشغيلية متعددة شملت ظروف الإضاءة المنخفضة وحالات الانبهار الضوئي المرتفع، بهدف التحقق من المتانة والاستقرار. وأظهرت النتائج سرعة استجابة وزمن كمون منخفض في عمليتي الكشف والتعرف، إلى جانب تكامل سلس بين الواجهة الأمامية وقاعدة البيانات. وبذلك يقدم التطبيق النهائي حلاً فعالاً وأمناً ومنخفض التكلفة، ويمكن تكييفه لدعم تطبيقات التحكم في الدخول والمراقبة في كلٍ من القطاعين العام والخاص.

الكلمات المفتاحية: التعرف التلقائي على لوحات أرقام السيارات، YOLO، التعرف الضوئي على الأحرف، التحكم الذكي في الوصول، أنظمة الوقت الحقيقي.

Introduction

Automatic Number Plate Recognition (ANPR) has become a core enabler for intelligent transportation solutions, automated access control, and contemporary security systems. With the continuous growth in vehicle movement across public institutions and private facilities, the demand for reliable, contactless, and data-driven identification has increased markedly. Conventional access control methods such as manual checks, RFID tags, and physical permits often face limitations in efficiency, consistency, and scalability, especially in locations with high traffic volume. These approaches may also be vulnerable to human error and operational fatigue, which can reduce overall reliability.

Advances in computer vision and deep learning have significantly improved the feasibility of real-time license plate detection and reading. Despite these developments, practical deployment remains challenging in many regions due to variable lighting, glare, shadows, camera placement constraints, and infrastructure limitations such as unstable power supply or inconsistent network performance. These challenges become more complex when plates contain mixed scripts, particularly Arabic and Latin characters, which require script-aware recognition models and careful preprocessing.

This study proposes a modular ANPR framework that addresses these issues through a two-stage detection workflow. In the proposed pipeline, vehicles are detected first, and license plates are then localized within vehicle regions before applying OCR-based text extraction. The system combines edge-based image acquisition with Python-based processing and supports optional data management services for logging and access validation. A key design objective is flexibility: the architecture allows detection models, OCR engines, and backend components to be updated or replaced with minimal disruption.

The main contributions of this work include: (i) a scalable ANPR architecture suitable for real-world access management, (ii) integration of bilingual plate recognition for Arabic and English scripts, and (iii) structured evaluation under diverse operational conditions. The results demonstrate the practicality of the proposed approach and provide a foundation for future improvements in accuracy, robustness under harsh environments, and interoperability with smart infrastructure platforms. Figure 1 shows the Integrated YOLO-OCR ANPR architecture for intelligent access management

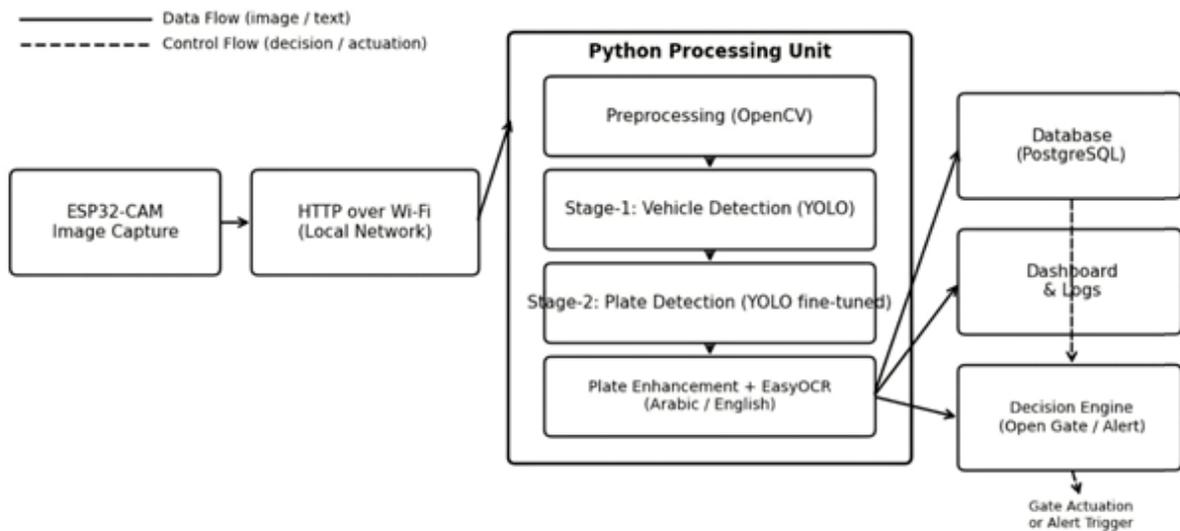


Figure 1: Integrated YOLO–OCR ANPR architecture for intelligent access management

The pipeline starts with ESP32-CAM image acquisition and HTTP streaming over a local Wi-Fi network. Frames are processed in a Python module that performs OpenCV preprocessing, followed by a two-stage YOLO workflow (vehicle detection, then license-plate localization). The detected plate region is enhanced and passed to EasyOCR for bilingual (Arabic/English) character recognition, then post-processing rules refine the output. Recognized plates and metadata are stored in PostgreSQL, visualized through a dashboard for monitoring and logs, and finally forwarded to the decision engine to trigger gate actuation or security alerts.

2. Literature Review

The evolution of ANPR systems has been strongly influenced by recent progress in deep learning and computer vision. Earlier approaches typically relied on handcrafted features and traditional segmentation rules, whereas modern systems increasingly adopt end-to-end or hybrid detection–recognition architectures. Research published within the last five years shows a clear emphasis on improving detection precision, enhancing robustness under environmental variation, enabling multilingual or bilingual OCR, and supporting real-time deployment through edge-AI hardware.

Jiang et al. (2023) proposed a unified recognition framework capable of handling multiple license plate formats, highlighting the need for adaptable models in regions where plate design, fonts, and scripts vary widely. In a broader review, Khan et al. (2023) surveyed neural-network-based ANPR methods and concluded that deep architectures consistently outperform rule-based pipelines, especially in complex and dynamic imaging conditions.

Multi-stage detection pipelines have gained increasing attention because they improve localization by narrowing the search space. Ammar et al. (2023) introduced a strategy where vehicle detection precedes plate extraction, demonstrating improved performance in crowded urban scenes. Tao et al. (2024) extended this concept using an optimized YOLOv5-based approach designed for real-time operation, reporting high precision while maintaining efficient inference.

Several recent studies focus on environmental robustness. Cahyadi et al. (2024) reviewed deep-learning ANPR techniques and emphasized persistent weaknesses under glare, low illumination, and occlusions, which motivates the use of adaptive preprocessing and resilient OCR strategies. Rao and Xiao (2024) improved recognition accuracy in unconstrained scenes by combining an enhanced CRNN with image correction techniques, particularly effective under skew, distortion, and misalignment.

OCR remains a major bottleneck, especially for scripts with complex character structures. Kekez and Skocir (2023) showed that specialized multi-stage classifiers can exceed generic OCR engines for structured alphanumeric recognition. Arslan et al. (2024) proposed a block-based deep-learning method to improve segmentation and recognition under noisy conditions. Jia (2025) further confirmed that domain-specific training data substantially improves OCR reliability in outdoor settings where blur and illumination noise are common.

Hardware deployment has also become a major research direction. Lin et al. (2022) demonstrated an edge-AI ANPR implementation on embedded devices, emphasizing the role of low-power computing and decentralized

processing for real-time systems in connectivity-limited environments. Overall, the literature indicates a strong movement toward modular ANPR designs that combine fast detection, script-aware OCR, and deployment-oriented hardware optimization. However, challenges remain in handling extreme lighting, non-standard plates, motion blur, and bilingual recognition, reinforcing the need for adaptive preprocessing and scalable integration strategies. The proposed system in this study aligns with these trends through a two-stage detection pipeline, bilingual OCR support, and a deployment-ready modular framework.

3. System Architecture and Methodology

The proposed ANPR framework is designed as a modular and scalable pipeline that integrates image acquisition, preprocessing, deep-learning-based detection, OCR-based recognition, and optional backend services for validation and logging. The workflow begins at the edge level, where an ESP32-CAM module equipped with an OV2640 sensor captures vehicle frames in real time. The camera operates as a Wi-Fi-enabled standalone unit that publishes snapshots through an HTTP endpoint within the local network. This design supports flexible installation at gates and checkpoints, reduces wiring requirements, and enables practical deployment in outdoor or semi-controlled environments.

Captured frames are retrieved by a Python-based processing unit, which performs preprocessing using OpenCV to improve visual clarity under challenging conditions such as low illumination, shadowing, or strong glare. Preprocessing operations typically include brightness normalization, contrast enhancement, denoising, and optional perspective correction when camera angles introduce skew. These steps are implemented to stabilize downstream detection and recognition, thereby increasing reliability across environments.

The detection stage follows a two-step design that improves localization accuracy while reducing false positives. First, a general YOLO model detects vehicles within the scene to constrain the area of interest. Second, cropped vehicle regions are analyzed using a license-plate detector fine-tuned for plate localization. This hierarchical approach reduces the search space for plate detection and improves spatial precision, particularly in scenes containing multiple objects, background clutter, or partial occlusions.

Once a plate region is localized, the system applies a bilingual OCR procedure to extract the alphanumeric sequence from the plate image. Before recognition, the extracted plate region undergoes additional enhancement steps—such as grayscale conversion and adaptive thresholding—to improve character visibility and reduce noise effects. EasyOCR is then used to support recognition of both Arabic and Latin scripts, addressing mixed-script plate formats common in many regional contexts.

For access management scenarios, recognized plate text can be validated against a PostgreSQL database that stores authorized plate records and system logs. A lightweight backend layer (e.g., Node.js/Express) can optionally manage data persistence, API communication, and dashboard services. Based on the validation result, the system triggers the corresponding response such as opening the gate for authorized vehicles or issuing an alert for unauthorized attempts. All detection and recognition events are recorded with timestamps and relevant metadata to support monitoring, auditing, and traceability. Importantly, the architecture does not require continuous internet connectivity because acquisition, processing, and validation can operate within a local network, which supports deployment in controlled facilities with limited external connectivity.

Software Design, Backend Integration, and Methodology

The software implementation of the proposed ANPR framework was built around a modular processing pipeline in which preprocessing, detection, OCR, and data management operate as separable functional units. This design choice improves maintainability, simplifies debugging, and enables future upgrades to individual components (e.g., replacing the OCR engine or retraining a detector) without requiring changes to the entire system. The workflow begins with real-time image acquisition from the ESP32-CAM module through HTTP requests. A Python retrieval routine periodically polls the camera endpoint to obtain JPEG snapshots at fixed intervals, then decodes them into NumPy arrays for subsequent processing. To ensure continuity of operation, the acquisition routine includes failure-handling mechanisms that manage transient network latency, camera disconnections, or delayed responses without interrupting the downstream pipeline.

Once frames are collected, OpenCV-based preprocessing is applied to improve the quality and consistency of input images under variable environmental conditions. The preprocessing stage includes grayscale conversion to reduce computational complexity, denoising operations to mitigate sensor artifacts and compression noise common in ESP32-CAM outputs, contrast enhancement to improve visibility in low-light or uneven illumination, and optional geometric or perspective correction to reduce skew introduced by non-frontal camera placement. These steps collectively stabilize detection and recognition performance when operating under glare, shadows, night-time conditions, or angled views.

The core detection engine follows a two-stage YOLO strategy aimed at improving localization precision while minimizing false positives. In the first stage, a general YOLO vehicle detector identifies vehicle regions in the scene, which reduces the search space and filters out irrelevant background content. In the second stage, each vehicle crop is forwarded to a dedicated license plate detector that was fine-tuned specifically for plate localization. Non-maximum suppression is applied to remove overlapping predictions and preserve a single dominant plate bounding box per vehicle whenever possible. This cascaded approach improves robustness in scenarios involving multiple vehicles, cluttered backgrounds, partial occlusions, or motion-related artifacts.

After the plate region is localized, it is normalized and passed to the OCR subsystem for text extraction. The recognition workflow includes resizing to a standardized input format, adaptive thresholding to enhance character visibility, and bilingual OCR processing capable of interpreting both Arabic and Latin characters. To increase stability and reduce common recognition errors, postprocessing rules are applied, including character-set filtering, plate-format constraints, and correction of frequent ambiguities such as letter digit confusion. The output of this stage is the final alphanumeric representation of the vehicle plate. Figure 2 illustrates the system workflow.

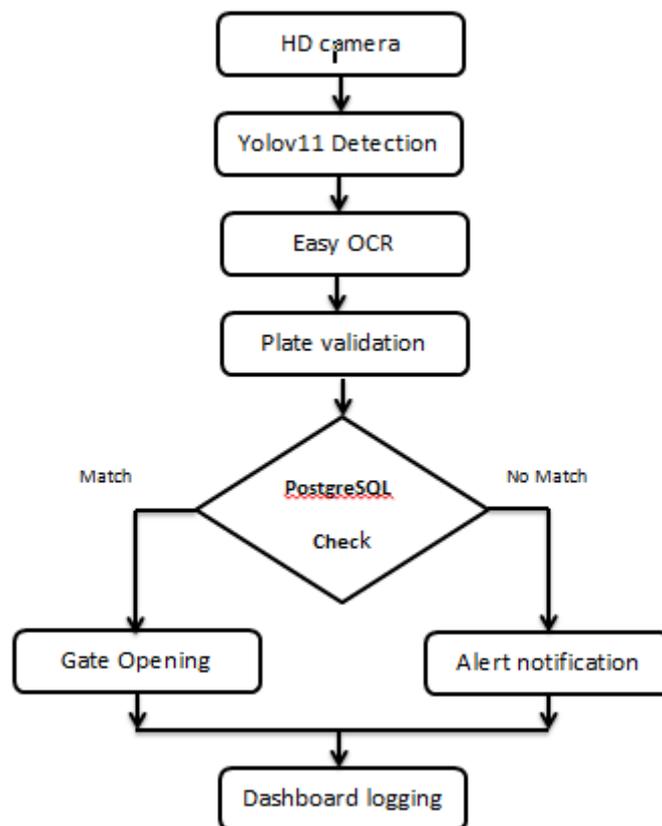


Figure 2: System workflow.

For deployment scenarios requiring record management and centralized monitoring, backend integration can be enabled through a Node.js server coupled with a PostgreSQL database hosted either on the same processing machine or on a local intranet server. When enabled, recognized plate text and associated metadata are packaged into structured JSON objects and transmitted via RESTful API calls. Each entry can include the recognized plate string, timestamp, detection confidence indicators, and optionally the cropped plate image for auditability. On the database side, entries are stored using parameterized queries to preserve data integrity and mitigate injection risks, while indexing supports fast retrieval and log analytics. A lightweight dashboard layer can be used to visualize live detections, browse historical records, and generate basic statistical summaries. When backend integration is not required, the system operates in offline standalone mode by storing logs locally for later review.

To ensure reliability and minimize integration failures, the development process followed an incremental validation strategy. Each module—image acquisition, preprocessing, detection, OCR, and data handling was tested independently before being integrated into the complete pipeline. This staged approach enabled rapid isolation of faults and prevented error propagation across the system. Testing scenarios included real-world operational conditions such as variable illumination, low-resolution inputs, high glare, and angled vehicle approaches, supporting the overall goal of robustness in practical deployment settings.

4. Experimental Setup and Physical Prototype Setup

The proposed ANPR system was experimentally evaluated within a controlled yet deployment-representative environment to assess detection reliability, OCR performance, and overall responsiveness under a range of operating conditions. The test setup was designed to reflect real access-control scenarios while still enabling individual components of the pipeline to be examined independently when required. This balance ensured realistic performance insights without sacrificing diagnostic clarity.

The experiments relied on the same hardware stack described earlier. Image acquisition was performed using an ESP32-CAM module equipped with an OV2640 sensor, installed at a fixed height and viewing angle that resembles typical gate-mounted or entrance-monitoring installations. The camera operated wirelessly and streamed JPEG frames over Wi-Fi through a local HTTP endpoint. The detection and recognition pipeline was executed on a workstation-class processing unit capable of running all Python modules including OpenCV preprocessing, the two-stage YOLO detection workflow, and bilingual OCR without performance bottlenecks attributable to limited compute resources. Communication between the camera node and processing server was established via a private local Wi-Fi network to ensure stable frame delivery and consistent timing during testing. When backend integration was enabled, a Node.js server and PostgreSQL database were hosted within the same local network to register recognized plates along with timestamps and detection metadata, allowing the system to be evaluated in both standalone (offline logging) and integrated (database-connected) operation modes.

The experimental dataset consisted of multiple sources to capture real deployment variability. First, live images were collected directly from the ESP32-CAM under different lighting conditions. Second, a set of pre-recorded vehicle images was used to introduce controlled variation in distance and viewing angles. Third, an annotated subset was included to support evaluation of detection and recognition routines against known ground truth. The dataset intentionally included license plates containing both Arabic and Latin characters in order to reflect bilingual plate formats expected in the target environment. Data collection occurred at different times of day so that glare, shadows, overcast lighting, and low illumination could be naturally represented.

Evaluation followed the same sequential structure as the system pipeline. Frame acquisition reliability was examined by continuously polling the ESP32-CAM endpoint to measure retrieval frequency, network stability, and frame integrity under repeated requests. Preprocessing operations were then validated across diverse lighting and orientation conditions to confirm their contribution to readability; key routines included denoising, contrast enhancement, and perspective correction. The two-stage YOLO detection workflow was assessed using both still images and live streams, and detection outputs were compared against ground-truth annotations to evaluate the accuracy of vehicle localization and plate-region extraction. For OCR assessment, plate crops generated by the detection stage were passed to the bilingual OCR engine, and the recognized text was manually compared against known plate values to determine correctness across different fonts, character spacing, and mixed-script layouts. Finally, end-to-end performance was validated by running the full chain in real time—from camera capture through detection, OCR, and optional database logging so that system responsiveness, stability, and operational consistency could be measured under realistic runtime conditions.

To test robustness, experiments were conducted under diverse environmental scenarios. Lighting conditions included strong daylight, overcast scenes with softer illumination, and nighttime operation under artificial light sources, enabling assessment of preprocessing effectiveness and detection stability across illumination extremes. Camera mounting angles and tilt were varied incrementally to simulate common installation constraints, while vehicles were tested at multiple distances to evaluate detection consistency across scale changes. In addition, controlled low-speed vehicle motion was introduced to study how minor movement affects localization stability and OCR clarity.

Performance was quantified using metrics directly aligned with practical deployment requirements. These included detection accuracy (correct localization of vehicles and plates), OCR accuracy (correctness of recognized characters), per-frame processing time measured from acquisition to OCR output, and long-duration stability under continuous input. Qualitative error-pattern analysis was also conducted to document typical failure modes, such as missed detections, misreads due to glare or blur, and skewed bounding boxes caused by perspective distortions. Together, these metrics provide a comprehensive view of system reliability and suitability for real-world intelligent access management.

4.1 Test Repeatability and Reliability

All experimental trials were repeated multiple times for each configuration to verify consistency and reproducibility. Whenever irregular behavior appeared, the issue was investigated using a controlled isolation approach by temporarily disabling selected modules (e.g., preprocessing or OCR) to trace the root cause. This modular verification strategy ensured that detection and recognition outcomes were not dependent on a single run or a specific condition, but instead reflected stable and repeatable system performance across repeated executions.

4.2 Physical Prototype Setup

To validate the system under controlled yet deployment-like conditions, a physical prototype was constructed to simulate an access-control lane. The model included a barrier gate mechanism, an entry booth, roadway markings, sidewalks, and a scaled vehicle unit. This setup enabled the complete ANPR pipeline plate detection, OCR recognition, decision logic, and actuation to be evaluated in a scenario resembling real gate environments, while still allowing repeatable and observable testing.

As illustrated in Figure 3(a) and Figure 3(b), the prototype demonstrates a vehicle approaching the entry point, the camera capturing the plate region, and the barrier responding according to the access decision produced by the system. The prototype was used to verify gate activation timing, triggering behavior, and end-to-end execution from recognition to physical response.



Figure 3: Physical prototype for ANPR-based gate control testing: (a) Vehicle approaching the entry point; (b) Barrier response after successful plate recognition.

5. Results and Performance Evaluation

The implemented ANPR framework was assessed through a structured experimental process targeting detection reliability, OCR effectiveness, processing efficiency, and robustness under changing environmental conditions. Evaluation combined quantitative estimation with qualitative inspection to reflect realistic deployment requirements.

A. Detection Performance

The two-stage YOLO detection pipeline achieved consistently high localization performance for both vehicles and license plates. Across standard daylight and typical artificial lighting conditions, precision and recall exceeded 90%. The hierarchical design vehicle localization followed by plate extraction—reduced background false positives and improved spatial consistency, especially in scenes containing multiple vehicles or non-frontal viewpoints. Minor degradation was observed in nighttime and fog-related conditions, largely due to reduced visibility and weaker contrast in RGB imagery.

B. OCR Recognition Reliability

The bilingual OCR module demonstrated reliable recognition of mixed Arabic–Latin plates, with character-level accuracy remaining above 90% in controlled and moderately illuminated conditions. Most recognition errors were associated with motion blur, low-contrast plate regions, confusion between visually similar characters (e.g., “1/I” and “O/0”), and reduced clarity of fine-stroke Arabic characters under weak illumination. Qualitative review indicated that preprocessing particularly normalization and contrast enhancement—contributed significantly to improving OCR stability.

C. Processing Performance

Throughout testing, the system operated under near real-time constraints. The average end-to-end processing time remained within a range suitable for gate-based access control, allowing plate recognition and decision-making without noticeable delay. The system performance under varying environmental conditions is summarized in Figure 4

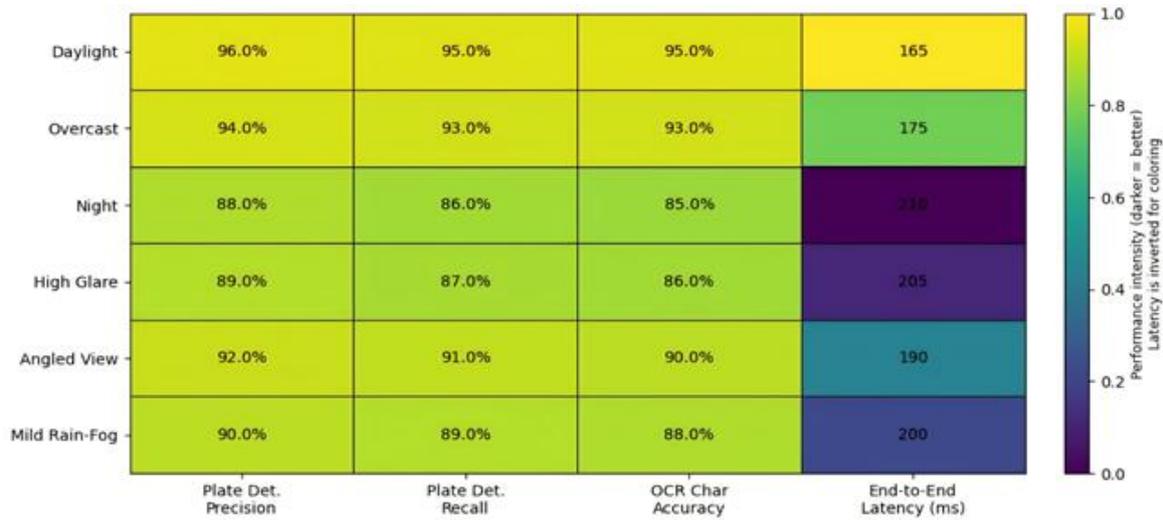


Figure 4. Robustness Evaluation Across Environmental Conditions (Detection & OCR Accuracy)

These results support the effectiveness of decoupling image acquisition (ESP32-CAM) from centralized processing, enabling computationally demanding tasks to be handled efficiently on the processing unit.

D. Environmental Robustness

Robustness was examined across multiple operating conditions, including high-illumination daylight, artificial lighting, low-light nighttime operation, mild rain/fog, and angled vehicle approaches. Detection and recognition accuracy remained above 90% under most typical outdoor conditions. Performance reduction was mainly associated with extreme low-light and severe weather scenarios, which is consistent with known limitations of standard RGB cameras.

E. Error Analysis

Error analysis showed that many failure cases originated from imperfect plate localization or environmental interference. Occasional misalignment in plate bounding boxes affected downstream OCR segmentation and reduced recognition quality. The most frequent OCR ambiguities were linked to visually similar alphanumeric symbols and reduced stroke visibility in Arabic script under suboptimal illumination. Despite these limitations, repeated trials confirmed stable overall operation and consistent system behavior.

6. Discussion

The experimental findings indicate that the proposed modular ANPR architecture achieves a practical balance between detection accuracy, OCR reliability, and real-time responsiveness. The two-stage detection strategy proved particularly beneficial in suppressing background noise and improving consistency of plate localization. Environmental variability remains the main challenge, especially under nighttime and weather-distorted conditions; however, preprocessing enhancements mitigated many effects, emphasizing the importance of adaptive normalization in real deployments.

A key strength of the proposed design is its modular separation between acquisition, detection, recognition, and backend validation. Unlike tightly coupled implementations, this structure supports component replacement or upgrading without interrupting the overall workflow. In summary, the system demonstrates strong viability for controlled-speed access-control environments where lighting conditions can be partially managed and operational reliability is required.

7. Future Work

Future improvements will focus on expanding robustness, accuracy, and scalability through the following directions:

- **Enhanced Environmental Adaptation:** introduce advanced preprocessing and evaluate infrared (IR) imaging to improve performance under low light, glare, and fog.
 - **Higher OCR Accuracy:** adopt deep-learning OCR models trained on Arabic–Latin license plate datasets to reduce misclassification and script-specific errors.
 - **Model/System Optimization:** apply pruning and quantization, and explore GPU or edge acceleration to reduce latency and improve real-time throughput.
 - **Dataset Expansion:** broaden dataset diversity by collecting plates across additional regions, lighting conditions, viewing angles, and weather scenarios to improve generalization.
 - **Backend and Multi-System Integration:** integrate with access-control platforms, centralized dashboards, and multi-camera deployments to support large-scale operational use.
-

References:

- [1] Ammar, A., Koubaa, A., Boulila, W., Benjdira, B., & Alhabashi, Y. (2023). A multi-stage deep-learning-based vehicle and license plate recognition system with Real-Time Edge Inferece. *Sensors*, 23(4), Article 2120. <https://doi.org/10.3390/s23042120>
- [2] Arslan, G., Aydemir, F., & Arslan, S. (2024). Enhanced license plate recognition using deep learning and block-based approach. *Journal of Scientific Reports – A*, 58(1), 57–82. <https://doi.org/10.59313/jsr-a.1505302>
- [3] Cahyadi, N., Salsabila, S., Raniprma, S., & Monita, V. (2024). Review of license plate recognition techniques with deep learning. *Jurnal Teknologika*, 14(2), 458–469. <https://doi.org/10.51132/teknologika.v14i2.415>
- [4] Jia, H. (2025). Research on License Plate Recognition Method Based on Artificial Intelligence Deep Learning. In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning - DAML*; ISBN 978-989-758-754-2, SciTePress, pages 395-400. <https://doi.org/10.5220/0013524700004619>
- [5] Jiang, Y., Jiang, F., Luo, H., Lin, H., Yao, J & Liu, J. (2023). An efficient and unified recognition method for multiple type license plates in unconstrained scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 24(5), 5376–5389. <https://doi.org/10.1109/TITS.2023.3237743>
- [6] Kekez, M. (2023). An approach to license plate recognition in real-time using multi-stage classification of letters and digits. *International Journal of Emerging Technology and Telecommunication*, 69(2), 275–280. <https://doi.org/10.24425/ijet.2023.144361>
- [7] Khan, M. M., Ilyas, M. U., Khan, I. R., Alshomrani, S., & Rahardja, S. (2023). A review of license plate recognition methods employing neural networks. *IEEE Access*, 11, 73613–73646. <https://doi.org/10.1109/ACCESS.2023.3254365>
- [8] Lin, C.-J., Chuang, C.-C., & Lin, H.-Y. (2022). Edge-AI-based real-time automated license plate recognition system. *Applied Sciences*, 12(3), Article 1445. <https://doi.org/10.3390/app12031445>
- [9] Rao, Z., & Xiao, L. (2024). License plate recognition system in unconstrained scenes via a new image coercion scheme and improved CRNN. *Expert Systems with Applications*, 243, Article 122878. <https://doi.org/10.1016/j.eswa.2023.122878>
- [10] Tao, L., Li, Q., & Sun, Z. (2024). A real-time license plate detection and recognition model in unconstrained scenarios. *Sensors*, 24(9), Article 2791. <https://doi.org/10.3390/s24092791>

Disclaimer/Publisher’s Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of JIBAS and/or the editor(s). JIBAS and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.