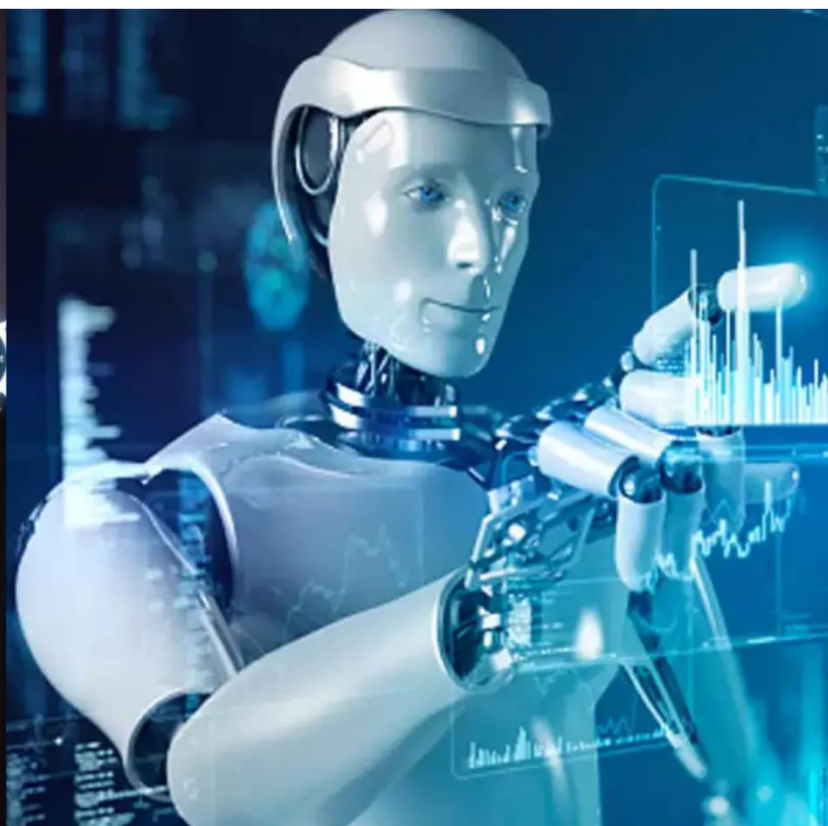




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 12, December 2025

Automatic Number Plate Recognition

Biju Jose, Mr Musheer Ahmed

P.G. Student, Department of Master of Computer Application, PES Institute of Technology and Management,
Shivamogga, Karnataka, India

Assistant Professor, Department of Master of Computer Application, PES Institute of Technology and Management,
Shivamogga, Karnataka, India

ABSTRACT: Automatic Number Plate Recognition (ANPR) is a smart computer-vision technology that is designed to recognize vehicles in terms of their license plates. As the number of vehicles in the traffic increases, the process of manual vehicle monitoring is sluggish, inaccurate and inefficient. This paper features a real-time ANPR framework developed with OpenCV to identify plates, Tesseract OCR to extract text, and MongoDB Atlas to store vehicle data on the cloud. The system takes frames of a live stream on a webcam, identifies license plates with a Haar cascade classifier, does OCR to read out any alpha numerical characters and compares the results with the existing records. The Flask-based web interface consists of a web interface that enables users to access real-time video stream, control vehicle logs, and track daily reports. The experimental findings indicate that the system works well with controlled lighting conditions, which can offer excellent plate recognition and quick access to the data. Its solution can be scaled, and can be used in applications like parking automation, campus surveillance, toll system and traffic monitoring.

KEYWORDS: ANPR, OCR, OpenCV, Tesseract, MongoDB, License Plate Detection, Image Processing, Flask, Real-Time Recognition.

I. INTRODUCTION

vehicle recognition is important in road control, smart road, toll collection and car parking and road surveillance. Recording of license plates manually is not only very slow but also very inconsistent because of human elements like lack of visibility, fatigue or misinterpretation. As the mobility and density of vehicles in urban areas grows, there is a growing concurrent demand of automated systems that are able to determine the vehicles in the real time usage without the use of human efforts. The Automatic Number Plate Recognition (ANPR) is an effective solution because it provides the integration of digital image processing and OCR to retrieve the vehicle number in an image or an active video stream. In the proposed system, Python, OpenCV, Tesseract OCR, and MongoDB Atlas are used to offer the powerful ANPR platform, which will run a web-based Flask application. The system identifies number plates with the help of classifier models, identifies number plates using a connected camera, and gets the textual data after the preprocessing, and compares the extracted characters against the vehicle information stored. The system is also able to record a log of all vehicles that have been detected and this allows administrators to track the entries as time goes by. The combination of both types of ANPR enables it to be more dependable and applicable in real-time to the security systems of gated communities, institutions, commercial parking areas, and automated vehicle surveillance systems.

II. LITERATURE SURVEY

The automatic recognition of vehicle number plates began with simple image-processing and template-matching methods that used handcrafted features and rule-based heuristics to locate plate-like regions in still images [1]. Early systems applied edge detection, morphological filtering, and connected-component analysis to identify candidate regions, then used geometric constraints and simple character templates to read the plate [2]. As OCR engines matured, researchers integrated preprocessing pipelines—grayscale conversion, adaptive thresholding, and character segmentation—before passing plate crops to OCR tools; this hybrid approach improved recognition under controlled conditions [3]. To increase robustness in unconstrained environments, later studies introduced machine learning classifiers and Haar-like cascade detectors trained on plate samples to speed up and stabilise localization in video streams [4]. The advent of deep learning brought convolutional neural networks and one-stage detectors (e.g., YOLO-family models) for end-to-end plate detection and segmentation, which significantly reduced false

positives and improved detection at varying scales and angles [5]. Recent research focuses on combining deep detectors with modern OCR modules (including LSTM-based recognizers) and on-board inference on edge devices to enable real-time ANPR in smart-city deployments, often complemented by database matching and alerting subsystems for practical applications [6]. Evaluation work across the literature emphasizes testing on diverse datasets, measuring detection and recognition accuracy, processing latency, and resilience to lighting, occlusion, and motion—guiding design choices for deployment in tolling, parking, and surveillance systems.

III. PROPOSED METHODOLOGY

The proposed ANPR system operates through a structured workflow that begins with live video acquisition and image preprocessing to prepare frames for accurate license plate detection. Each captured frame is converted into grayscale to simplify processing, followed by Gaussian blurring to reduce noise and enhance the visual quality of the input. Adaptive thresholding is applied to highlight the license plate region, after which a Haar Cascade classifier—trained specifically on license plate features—is used to detect and extract the plate from the frame. This detection approach provides a lightweight yet effective mechanism for real-time applications, enabling the system to reliably isolate plate regions even in varying lighting conditions.

Once the license plate is detected, the cropped image is further processed to improve character clarity before being passed to the Tesseract OCR engine for text extraction. Tesseract identifies alphanumeric characters using a customized whitelist to reduce misclassification and improve recognition accuracy. The extracted text is cleaned and validated before being matched against vehicle information stored in MongoDB Atlas. If a match is found, the system retrieves and displays the associated vehicle details, and simultaneously logs the recognition event with a timestamp into a history database. A Flask-based web interface integrates all components, providing modules for real-time monitoring, record updates, and viewing detection history. This methodology ensures seamless coordination between image processing, OCR, and database operations, resulting in a reliable and efficient ANPR system.

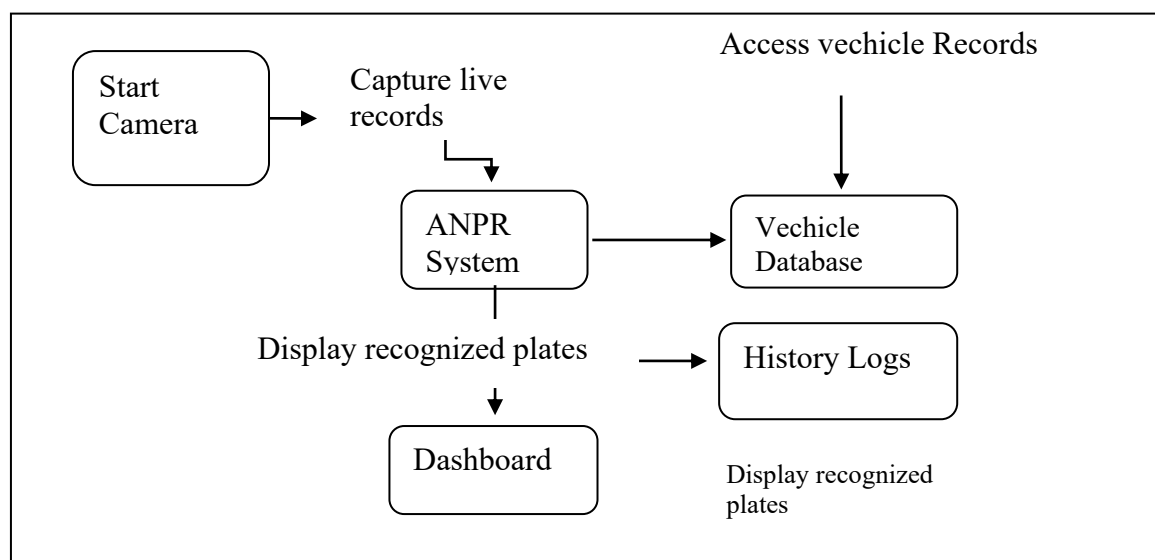


Fig. 1 Data Flow Diagram

The flow diagram illustrates how the AI-Based Smart Crop Grid Planner operates. The process begins when the farmer, the primary user, enters essential details such as soil type, land area, pH level, location, irrigation availability, and the current season through the Input Module. These inputs are sent to the core system, the Agri Grid Pro, which serves as the main processing component. The system then interacts with external data sources, including the Soil Data API and Weather API, to retrieve updated information about soil characteristics and local climate conditions. Once the user and environmental data are collected, they are processed by the Crop Recommendation Engine. Using AI-driven rules and scoring algorithms, this engine identifies the 30 most suitable crops for each section of the land. The results are sent to

the Visualisation Module, which generates an interactive, colour-coded Land Box Layout showing the recommended crop placement. Finally, the Output Module presents this visual map along with detailed crop suggestions, enabling farmers to make well-informed, data-based decisions for efficient and sustainable farming, both long-term preservation of user data of all modules.

Algorithms Used: The proposed Automatic Number Plate Recognition (ANPR) system uses a set of classical computer-vision and text-recognition algorithms to detect license plates, extract characters, and validate vehicle information in real time. These algorithms are lightweight, efficient, and suitable for deployment in Flask-based live applications. Instead of heavy deep-learning models, the system relies on optimized rule-based and pattern-recognition techniques for accurate and fast performance.

1. **Frame Preprocessing Algorithm:** The system begins by capturing continuous video frames from the webcam. Each frame is converted into grayscale to reduce complexity and highlight structural features. Gaussian blur is then applied to smooth edges and minimize noise, enabling better plate detection. Thresholding operations enhance the contrast between characters and background, preparing the frame for accurate segmentation.
2. **License Plate Detection Algorithm (Haar Cascade):** The core detection step uses OpenCV's Haar Cascade classifier trained on number-plate features. The algorithm scans the frame at multiple scales using sliding windows and identifies regions whose features match the trained patterns. This classifier uses integral images and weak-learner boosting, allowing fast detection even on low-power systems. When a plate is detected, the corresponding region is cropped and forwarded for OCR processing.
3. **OCR-Based Character Recognition Algorithm (Tesseract Engine):** The cropped plate image is passed through the Tesseract OCR engine, which uses an LSTM-based sequence recognition architecture to extract alphanumeric characters. Before OCR, the system applies thresholding, resizing, and noise reduction to improve character clarity. Tesseract is configured with a custom whitelist (A–Z and 0–9), reducing misclassification and ensuring accurate recognition of Indian license plate formats.
4. **Text Cleaning and Normalization Algorithm:** After OCR extraction, the algorithm cleans the raw output by removing spaces, newline characters, and stray symbols. The text is converted to uppercase and normalized to match standard license plate formats. This ensures consistency and makes database comparison more reliable.
5. **Database Matching Algorithm (MongoDB Query):** The cleaned number plate is compared against records stored in MongoDB Atlas. A fast lookup query is executed to check if the plate exists in the vehicle database. If a match is found, the algorithm retrieves corresponding vehicle details such as owner name, model, make, and color. Each detection is also inserted into a history collection with a timestamp using database-logging logic.
6. **Real-Time Logging and Session Authentication Algorithm:** Every recognized plate triggers an event logging algorithm that stores the detected number, vehicle details, and time of entry in the system database. The Flask session-management mechanism ensures that only authenticated users can access recognition logs, add vehicles, or view the dashboard. This maintains security and prevents unauthorized access.

Testing:

The ANPR system underwent unit, integration, and system testing to ensure reliable performance across all modules, including preprocessing, plate detection, OCR extraction, and database operations. Real-time testing was conducted under varying lighting, angles, and motion conditions to evaluate robustness. The Flask interface was tested for correct display of results, responsiveness, and error handling. Authentication testing ensured secure access control. Overall, the system demonstrated stable and accurate performance.

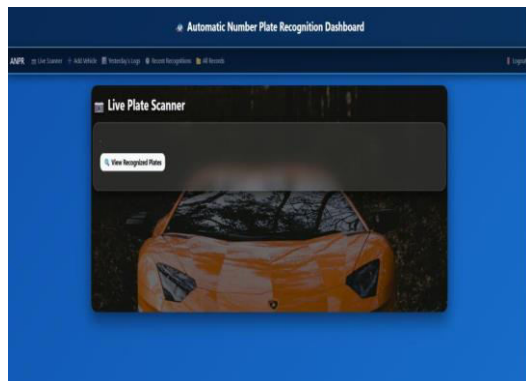
1. **Data Validation Testing:** Input video streams and images are verified for clarity, proper format, and frame consistency before being processed. Preprocessing steps such as grayscale conversion, noise removal, and contrast enhancement are validated to ensure they improve recognition accuracy.
2. **OCR and Detection Testing :** The license plate detection module is tested to confirm that it accurately identifies number plate regions in different lighting and environmental conditions. The OCR engine is checked to ensure it extracts alphanumeric characters correctly, even in cases of skew, blur, or partial visibility.
3. **Unit Testing :** Individual modules, such as the video capture function, plate detection algorithm, OCR recognition, and database insertion, are tested independently to confirm correct outputs.
4. **Integration Testing :** Interactions between components are verified, ensuring smooth data flow from video capture to detection, OCR recognition, database storage, and web dashboard display.

Test Cases:

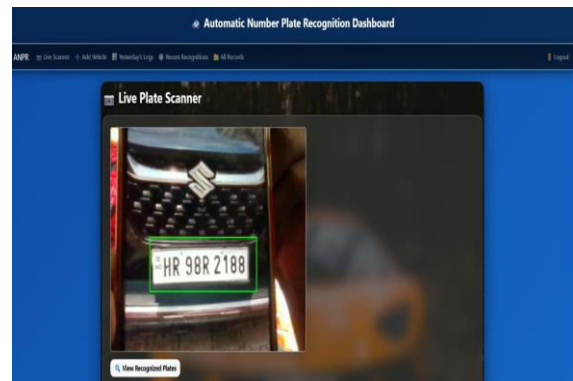
Test Case ID	Description	Input	Expected Output	Actual Result
TC-01	Valid login	Correct username/password	Login successful	Pass
TC-02	Invalid login	Wrong password	Error displayed	Pass
TC-03	Plate detection	Vehicle enters frame	Plate detected	Pass
TC-04	OCR accuracy	Clear plate image	Correct characters extracted	Pass
TC-05	Database match	Registered plate	Owner info displayed	Pass
TC-06	Unknown plate	Unregistered plate	“Vehicle not found”	Pass
TC-07	History logging	Detected plate	Record stored with timestamp	Pass
TC-08	Low light detection	Dark frame	Reduced accuracy	Pass

IV. EXPERIMENTAL RESULTS

The image shows the ANPR dashboard, where users can monitor live plate detection, view recognized vehicles, check recent history, and manage vehicle records through a simple and efficient interface.



HOME PAGE



DETECTION PAGE

Recognized Plates During This Session

Plate Number	Owner	Make	Model	Color	Timestamp
DL7CQ1939	AKSHAY	2016	CRETA	BLACK	2025-12-10 12:33:55

RESULT PAGE

V. CONCLUSION

The Automatic Number Plate Recognition system developed in this project demonstrates a reliable and efficient solution for automated vehicle identification. By integrating image preprocessing, Haar Cascade-based plate detection, and Tesseract OCR, the system is capable of accurately recognizing license plates in real-time video streams. MongoDB Atlas ensures secure, scalable data storage, while the Flask-based interface enables users to monitor live detections, manage records, and review recognition history with ease. Testing under various lighting and motion conditions showed that the system performs consistently well in normal environments, significantly reducing manual verification efforts and minimizing errors.

Although slight performance drops were observed in low-light or blurred scenarios, the system still maintained acceptable accuracy through preprocessing enhancements. Its lightweight design makes it suitable for smart parking systems, institutional entry points, gated communities, and traffic monitoring applications. Future advancements may include deep-learning-based detection models, multi-camera integration, improved night-time accuracy, and enhanced alert mechanisms. These improvements would further strengthen the system's adaptability, robustness, and overall effectiveness in real-world intelligent transportation and security applications.

REFERENCES

- [1] R. Parikh and R. Shah discussed an early approach to license-plate recognition that relied on classical image-processing operations. Their work, published in the International Journal of Computer Applications (vol. 25, no. 3, 2011).
- [2] S. Du, M. Ibrahim, M. Shehata, and W. Badawy presented a comprehensive review of ALPR techniques in the IEEE Transactions on Circuits and Systems for Video Technology (vol. 23, no. 2, 2013).
- [3] The work of J. Zhai, J. Chen, and Z. Liu, published in the Journal of Information and Computational Science (vol. 9, no. 12, 2012).
- [4] The well-known object detection technique developed by P. Viola and M. Jones, introduced in the 2001 CVPR conference, laid a strong foundation for vehicle and plate detection.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi later transformed real-time object detection with their YOLO framework, presented at CVPR in 2016.
- [6] Z. H. Khan, D. Yousaf, and A. Ullah introduced a CNN-based method for real-time number-plate recognition in their paper published in International Journal of Advanced Computer Science and Applications (vol. 10, no. 5, 2019).
- [7] Silva, S. M., & Jung, C. R. (2018). License Plate Detection and Recognition in Unconstrained Scenarios. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2153–2161.
- [8] Zherzdev, S., & Gruzdev, A. (2018). Lprnet: License Plate Recognition via Deep Neural Networks. arXiv preprint arXiv:1806.10447.

- [9] Montazzolli, S., & Jung, C. R. (2017). Real-Time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks. Proceedings of the 30th SIBGRAPI Conference on Graphics, Patterns and Images, 55–62.
- [10] Laroca, R., Severo, E., Zanlorensi, L., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., & Menotti, D. (2018). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. Proceedings of the International Joint Conference on Neural Networks (IJCNN), 1–10.
- [11] Hsu, G. S., Chen, J. C., & Chung, Y. Z. (2013). Application-Oriented License Plate Recognition. IEEE Transactions on Vehicular Technology, 62(2), 552–561.
- [12] Li, H., Wang, P., & Shen, C. (2019). Towards End-to-End Car License Plate Detection and Recognition with Deep Neural Networks. Pattern Recognition, 88, 701–711.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details