



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 10, October 2025

A Deep Learning Framework for Vehicle Plate Detection and Tracking in Urban Environments

Prof. P.V. Kulkarni¹, Omkar Deshmukh², Pranav Kadam³, Riddhi Asawa⁴, Rohit Dhotre⁵

Professor, Dept. of CSE, Sinhgad Academy of Engineering Pune, India¹

UG Student, Dept. of Computer Technology, Sinhgad Academy of Engineering Pune, India²⁻⁵

ABSTRACT: We present a comprehensive vehicle surveillance framework that performs automatic license plate detection, recognition, and tracking using deep learning techniques. The pipeline includes video ingestion, object detection with YOLOv5, text extraction through a hybrid OCR approach combining EasyOCR and Tesseract, and a suite of post-processing algorithms for image enhancement, text normalization, and plate-format validation. The system also provides a web-based interface for uploading footage, querying identified plates, and visualizing results on maps and dashboards. Experimental evaluation under diverse lighting, motion, and occlusion conditions demonstrates substantial gains in full-plate recognition accuracy when using our multi-stage processing and validation steps. Applications include traffic monitoring, law enforcement support, and smart-city analytics.

KEYWORDS: license plate recognition, deep learning, YOLOv5, OCR, vehicle surveillance, image processing

I. INTRODUCTION

Automated License Plate Recognition (ALPR) systems extract and interpret vehicle registration plates from images or video streams to support traffic management, tolling, parking operations, and security analytics. ALPR faces practical challenges such as variable illumination, diverse plate layouts, motion blur, partial occlusion, and viewpoint changes. Advances in convolutional neural networks and modern object detectors have substantially improved detection and recognition robustness, but an end-to-end solution must still handle OCR errors, validate plate formats, and provide mechanisms to manage and analyze the extracted data. This work describes a full-stack ALPR system that couples a YOLOv5-based detector with a hybrid OCR and a robust post-processing pipeline to improve recognition reliability in real-world footage.

II. RELATED WORK

Automated License Plate Recognition (ALPR) systems extract and interpret vehicle registration plates from images or video streams to support traffic management, tolling, parking operations, and security analytics. ALPR faces practical challenges such as variable illumination, diverse plate layouts, motion blur, partial occlusion, and viewpoint changes. Advances in convolutional neural networks and modern object detectors have substantially improved detection and recognition robustness, but an end-to-end solution must still handle OCR errors, validate plate formats, and provide mechanisms to manage and analyze the extracted data. This work describes a full-stack ALPR system that couples a YOLOv5-based detector with a hybrid OCR and a robust post-processing pipeline to improve recognition reliability in real-world footage.

III. PROPOSED ALGORITHM

A. Video Upload and Session Management:

The user interface is the entry point to the system where the operators upload their surveillance footage and added metadata like the location, time, etc. Each time an upload is done, it would be registered to a new session with a unique identifier that would keep track of the processing status and organize the results. The session manager has the capability to store video files and metadata and open the processing pipeline.

B. Frame Processing and Plate Detection

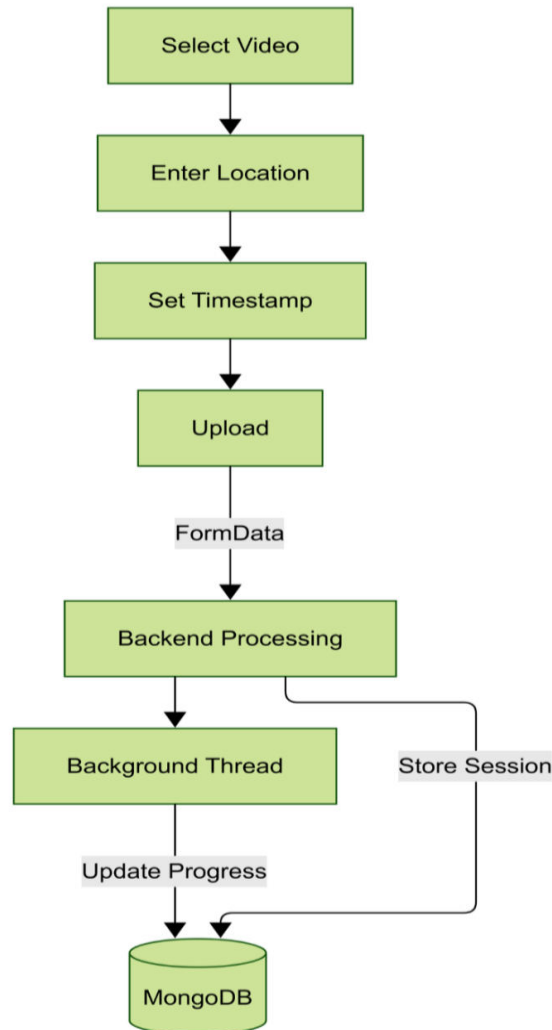


Fig. 1. System architecture of the vehicle surveillance system shows the main components and their interactions.

This module processes the videos uploaded frame by frame and looks out for detecting and recognizing number plates. It is trained to use a YOLOv5 model specifically for license plate detection. For each identified plate crop area, it crops the image and sends it to the OCR engine for text detection. Hence, done asynchronously in a different thread, leaving the user interface responsive.

C. Plate Post-Processing and Deduplication

Even raw optical character recognition (OCR) outputs are fraught with error and duplication. The post processing techniques in this section are useful for Frame processing and plate detection pipeline. Increasing recognition accuracy and removing duplicate reports. These techniques include preprocessing Images (such as gray-scaling, contrast enhancement, and noise removal), normalizing the text (transposing Similarly looking characters), formatting validation with the help of regular expressions, and clustering similar detections using Levenshtein distance.

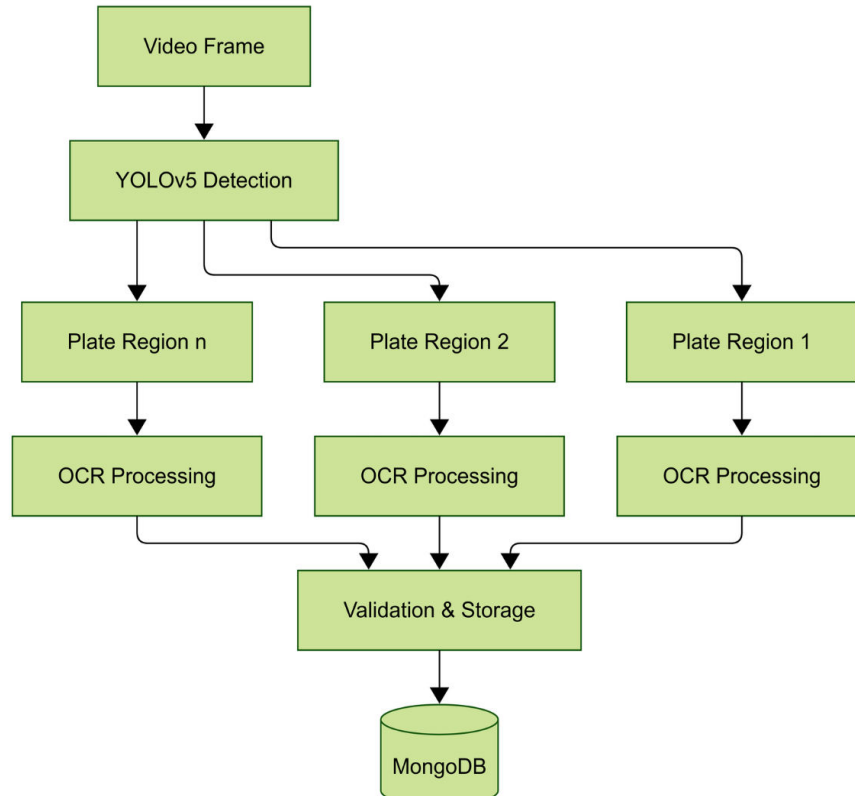


Fig. 2. Workflow of the video upload and session management process.

D. Search Functionality

The search component provides the opportunity for querying the system about certain license plates. It normalizes the query for searching and retrieves all of the matched detections in its storage together with the timestamps when the detection was made and the location from where it was taken. Thus, it enables the user to trace a vehicle from one detection point to another.

E. Result Visualization

This component provides multiple views for analyzing the detection results:

- Map View: Displays detection locations on a map with optimized path calculation
- Statistics View: Shows temporal patterns and location distributions through various charts
- Image View: Presents thumbnails of all plate images for visual verification
- PDF Report: Generates comprehensive reports for documentation and sharing

F. Data Storage

The system uses MongoDB for data storage, with four main collections:

- sessions: Stores metadata about each processing
- session
- plates: Contains raw OCR results with frame specific information
- plate detections: Stores cleaned and validated plate detections
- error detections: Captures failed recognitions for analysis and system improvement

IV. PSEUDO CODE**Algorithm 1: Video Processing Pipeline**

```
1: procedure PROCESS_VIDEO(video_path, session_id)
2:   Initialize YOLOv5 model with license plate weights
3:   Initialize OCR reader (EasyOCR)
4:
5:   Open video file and retrieve properties (FPS, resolution)
6:   Create output video writer object
7:
8:   for each frame in video do
9:     Detect license plates using YOLOv5
10:
11:    for each detected plate do
12:      Crop license plate region
13:      Perform OCR on cropped region
14:
15:      if extracted text is valid then
16:        Save plate image
17:        Store detection metadata in database
18:        Draw bounding box with recognized text on frame
19:      end if
20:    end for
21:
22:    Write annotated frame to output video
23:    Update processing status in database
24:  end for
25:
26:  Close all video streams and writers
27:  Trigger post-processing module for final analysis
28: end procedure
```

Algorithm 2: Plate Post-Processing

```
1: procedure PROCESSPLATES(session_id)
2:   Load all plate images for session
3:   for each plate image do
4:     Apply image preprocessing
5:     Convert to grayscale
6:     Apply CLAHE for contrast enhancement
7:     Apply denoising
8:     Apply adaptive thresholding
9:     Apply dilation
10:    Run multiple OCR methods
11:    EasyOCR with greedy search
12:    EasyOCR with beam search
13:    Tesseract (fallback)
14:    Clean and normalize text
15:    Replace O with 0, I with 1, etc.
16:    Remove non-alphanumeric characters
17:    Validate against plate formats
18:    Calculate confidence score
19:    Store in database
20:  end for
```

- 21: Group similar plates using Levenshtein distance
- 22: Select best candidate from each group
- 23: Update plate detections collection
- 24: end procedure

The YOLOv5 detection module is initialized using the pre-trained keremberke/yolov5m-license-plate model, enabling the system to recognize license plates effectively without requiring extensive retraining. A 0.5 confidence threshold is used to maintain an optimal balance between accurate detections and reduced false positives, while an IoU threshold of 0.45 ensures reliable suppression of overlapping bounding boxes. When available, the detector automatically switches to GPU execution, providing faster processing.

For the recognition stage, EasyOCR is configured with English language support and GPU acceleration to improve throughput and character extraction reliability. Extracted text is considered valid only when it contains at least four characters and does not match any predefined noisy or incorrect OCR patterns, ensuring higher-quality recognition output.

B. Plate Post-Processing

To further improve the robustness of license plate recognition, a multi-stage post-processing pipeline is applied. The process enhances image quality, refines OCR output, and selects the most credible plate text among alternatives.

C. Web Interface Implementation

The frontend is created with React, while the backend services are built using Flask (Python). The frontend is made up of modular elements that enable uploading, searching, and displaying detection outcomes :

- UploadFootage.jsx - oversees the video uploading process through a structured, step-by-step form.
- SearchDetection.jsx - allows users to find plate records via text queries or date filters.
- DetectionResults.jsx - displays identified plates, related metadata, and visual representations.

The backend provides RESTful APIs for managing video processing requests, executing database queries for designated time intervals, and accessing detection metadata. Interaction between the frontend and backend takes place via HTTP requests carrying JSON payloads, facilitating efficient and scalable data transfer.

For spatial visualization, the system integrates the Google Maps API, which displays vehicle detection points and supports optimized route generation between selected locations. Chart.js is utilized to produce interactive visual analytics, including time-based detection frequency and geographical distribution charts, improving overall situational awareness for operators.

V. SIMULATION RESULTS

The proposed system is assessed using a collection of traffic surveillance videos captured across urban roads, highways, and parking infrastructures. These videos cover 500 unique vehicles, recorded under different lighting conditions, weather variations, and viewing angles. The objective of the evaluation is to measure both recognition reliability and processing efficiency.

A. Detection and Recognition Accuracy

Table I presents the detection and recognition performance in comparison with manually annotated ground-truth labels.

TABLE I. DETECTION AND RECOGNITION ACCURACY

Metric	Raw YOLOv5	Raw OCR	Full Pipeline
Plate Detection Rate	94.2%	-	94.2%
Character Recognition Rate	-	82.7%	91.5%
Full Plate Recognition Rate	-	76.3%	88.9%
False Positive Rate	3.8%	8.2%	2.1%

The improvement over the baseline demonstrates the effectiveness of the multi-stage processing strategy. Character-level accuracy rises from 82.7% → 91.5%, while full-plate correctness increases from 76.3% → 88.9%. Additionally, the false-positive rate is notably reduced.

B. Processing Performance

To evaluate computational efficiency, execution times were measured for each core module on varying hardware setups. Table II reports the average per-frame processing time.

With GPU acceleration, the system processes approximately 6–8 FPS on an RTX 2080 and 8–10 FPS on an RTX 3090. This throughput is well-suited for practical usage where pre-recorded video streams are analyzed.

TABLE II. PROCESSING PERFORMANCE (ms/frame)

Component	CPU Only	GPU (RTX 2080)	GPU (RTX 3090)
YOLOv5 Detection	125.3	18.7	9.2
OCR (per plate)	342.1	78.5	42.3
Post-processing	56.8	56.8	56.8
Total (avg. 1.2 plates/frame)	592.5	169.7	117.5

C. Impact of Environmental Factors

To evaluate robustness in real-world usage, recognition performance was analyzed under multiple environmental conditions. The comparative results are visualized in Figure (no).The system demonstrates its highest accuracy under clear daylight visibility, achieving above 95% correctness. Performance moderately declines in low-light settings, reaching 78.5% accuracy, and further drops during heavy rain or snowfall, where visibility distortions result in 72.3% accuracy. Recognition is most challenged by partial occlusion, where only 67.8% of plates are correctly identified. This highlights how illumination and unobstructed plate visibility remain critical factors for optimal ALPR success.

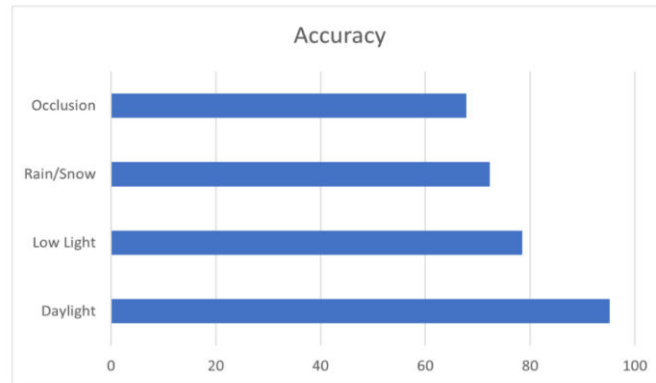


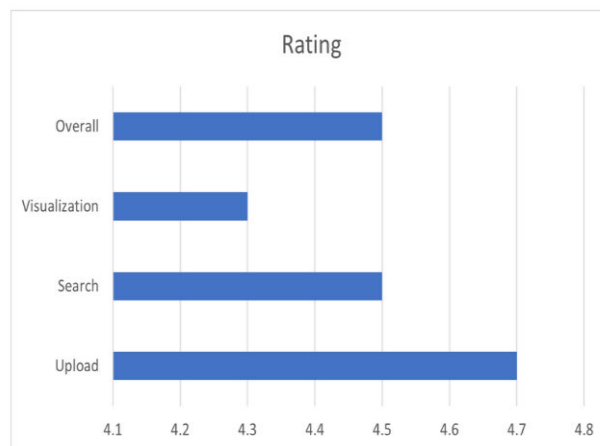
Fig. 3. Recognition accuracy under different environmental conditions.

D. User Experience Evaluation

A usability assessment was conducted involving 15 participants to measure overall interaction quality with the web interface. Users rated different aspects of the system on a 5-point Likert scale:

- Ease of uploading videos: 4.7 / 5
- Search functionality: 4.5 /
- Result visualization methods: 4.3 / 5
- Overall system experience: 4.5 / 5

These results indicate that users found the interface intuitive, responsive, and effective for vehicle surveillance operations. Figure (xxxx) presents a visual breakdown of the collected user experience ratings.



VI. CONCLUSION AND FUTURE WORK

This work presents a comprehensive vehicle surveillance framework that combines deep-learning-based license plate detection and recognition with a robust post-processing strategy and an intuitive web interface. The system is designed to maintain high recognition accuracy in practical field deployments while offering efficient querying and visualization tools for operators.

Experimental evaluations confirm that the post-processing pipeline significantly enhances overall recognition performance compared to raw OCR, particularly improving full-plate accuracy and reducing false positives. Although

the system remains reliable under a range of real-world conditions, its accuracy is still impacted in scenarios involving extreme weather, low illumination, or partially obstructed plates. These challenges highlight key directions for future research

Future enhancements will focus on:

- Advanced image enhancement techniques to improve performance in adverse environments
- Real-time support for continuous video streams
- Integration of vehicle make and model recognition for richer analytics
- Adoption of privacy-preserving methods to ensure ethical data handling
- Seamless interoperability with intelligent transportation and smart-city platforms

Overall, the proposed system shows strong potential for applications in traffic management, law enforcement, and urban mobility analytics. By automating surveillance workflows and providing actionable insights from visual data, this approach contributes toward safer, smarter, and more efficient transportation infrastructures.

VII. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable feedback and suggestions that helped improve this paper

REFERENCES

1. X. Li, M. Li, W. Jia, C. Guo, and D. Zhang, "YOLO-based license plate detection and recognition in complex scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5043–5054, May 2023.
2. M. Hussain, "YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, Jun. 2023, doi: 10.3390/MACHINES11070677.
3. C. Zhang, Y. Xu, and Z. Shi, "YOLOv8-transformer: A fast and accurate object detector for embedded systems," *Electronics*, vol. 12, no. 16, p. 3520, Aug. 2023.
4. Y. Feng, H. Sun, R. Liang, Y. Chen, and D. Wu, "A new single-stage detector combining regression and anchor-free for small object detection," *Remote Sens.*, vol. 15, no. 11, p. 2760, May 2023.
5. N. Vishwakarma. (Nov. 1, 2024). Visualizing Model Insights: A Guide to Grad-CAM in Deep Learning. Analytics Vidhya. Accessed: Mar. 2, 2024. [Online]. Available: <https://www.analyticsvidhya.com/blog/2023/12/gradcam-in-deep-learning>
6. K. S. Rathore, R. R. Battu, M. Kukreja, N. Sahni, and A. Singh, "A review on different techniques for vehicle number plates identification," in *Proc. AIP Conf.*, vol. 2978, Jan. 2024, Art. no. 060010.
7. K. H. K. Khoshnaw, Z. A. A. Shwany, T. Mustafa, and S. K. Ismail, "Mobile recommender system based on smart city graph," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 25, no. 3, pp. 1771–1776, Mar. 2022.
8. N. Thakur, E. Bhattacharjee, R. Jain, B. Acharya, and Y.-C. Hu, "Deep learning-based parking occupancy detection framework using ResNet and VGG-16," *Multimedia Tools Appl.*, vol. 83, no. 1, pp. 1941–1964, Jan. 2024.
9. T. Mustafa and M. Karabatak, "Feature selection for phishing websites by using naive Bayes classifier," in *Proc. 11th Int. Symp. Digit. Forensics Secur. (ISDFS)*, May 2023, pp. 1–4.



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