www.ijarp.com ISSN 2456-9992 Page: 01-08

International Journal Advanced Research Publications

Review Article

Volume: 02
Issue: 04

CONVOLUTIONAL NEURAL NETWORK-BASED AUTOMATIC LICENSE PLATE RECOGNITION FOR BETTER RESPONSES

***Suket Tyagi, Yash Sharma**

Department of Computer Science and Engineering, RD Engineering College, Ghaziabad, India.

Article Received: 17 March 2026, Article Revised: 07 April 2026, Published on: 27 April 2026

***Corresponding Author: Suket Tyagi**
Department of Computer Science and Engineering, RD Engineering College, Ghaziabad, India.
DOI: https://doi-doi.org/101555/ijarp.9317

ABSTRACT

The rapid growth of Intelligent Transportation Systems (ITS) and the increasing number of vehicles have created a strong demand for automated traffic monitoring and law enforcement solutions. Automatic License Plate Recognition Systems (ALPRS) play a vital role in applications such as traffic surveillance, toll collection, parking management, and vehicle tracking. However, traditional NPRS systems face significant challenges in real-world environments due to varying plate formats, motion blur, low lighting, and adverse weather conditions. This research proposes a deep learning-based Automatic License Plate Detection and Recognition (ALPDR) system that integrates YOLOv10 [14] for license plate detection and PaddleOCR [15] for character recognition. The system follows a multi-stage pipeline including image preprocessing, plate detection, character segmentation, and optical character recognition. The proposed approach enhances detection accuracy and real-time performance by combining convolutional neural networks with efficient OCR techniques. Experimental evaluation on Indian vehicle license plate images demonstrates the effectiveness of the system under varying environmental conditions. The proposed framework can be integrated with IoT-based smart transportation systems to enable automated traffic monitoring, vehicle identification, and intelligent law enforcement.

KEYWORDS: Number plate recognition system, YOLOv10 algorithm, Computer Vision, PaddleOCR, CNN.

INTRODUCTION

The Number Plate Recognition System (NPRS) is primarily used to extract details from the number plate attached to a particular vehicle. It is highly useful in traffic management systems and in automatic parking ticket generation. NPRS employs algorithms designed for high efficiency and accurate result computation. This system is capable of fetching the number plate details of vehicles, thereby eliminating the need for human intervention.

Processing plate images captured at various tilted angles or under noisy conditions makes recognition tasks considerably more difficult. The problem requires both high precision and rapid processing, as it is frequently deployed in real-time systems. Most Vehicle License Plate (VLP) recognition applications constrain the position, tilt angles, and distance between the camera and vehicles in order to simplify the recognition task. Consequently, the recognition performance of VLP systems has improved significantly. Additionally, accuracy can be further increased by exploiting specific characteristics of local VLPs, such as the number of rows on a plate, character count, the background colour, or the width-to-height ratio of the plate [1].

Fig1: Samples of vehicles captured by camera.

Related Work

Research on automatic number plate recognition originated in the 1990s. Early approaches were based on the properties of boundary lines: images were processed using gradient filters to highlight edges, binarized, and then analyzed using algorithms such as the Hough transform to detect lines. Pairs of parallel lines were subsequently considered plate candidates [4][5]. Another category of methods relied on morphological image processing, exploiting properties such as brightness, symmetry, and angular orientation of plate regions [2].

License plate detection systems are designed to identify patterns and features characteristic of license plates, enabling reliable localization within larger images. One notable approach involves analyzing the texture patterns and border characteristics of candidate regions. Images are scanned using multi-scale sliding windows, and each candidate region is verified by a classifier trained to distinguish actual license plates from visually similar backgrounds. This approach was adapted from text-detection methods and extended specifically for license plate localization. A number of additional methods have also been proposed for VLP detection from video data, where objects appear as sequences in consecutive image frames [3][6].

METHODOLOGY

The proposed system follows a multi-stage processing pipeline comprising four main stages: image preprocessing, registration plate detection, image breakdown (segmentation), and optical character recognition using PaddleOCR [15].

A. Image Preprocessing

Camera images are first processed to enhance edges, which assists the system in locating license plates more accurately. The images are converted to grayscale and normalized, and contrast is adjusted. Sobel filters are then applied for edge detection, and the results are binarized using an adaptive thresholding method chosen for computational efficiency. These preprocessed images are forwarded to the license plate recognition pipeline.

B. Registration Plate Detection

In the boundary-based approach, edge detection is the most critical step. The Hough transform is applied to the binary image to identify lines from object images; pairs of parallel lines define the candidate plate region. However, a known limitation of this method is the high computational cost of the Hough transform when applied to binary images with a large number of pixels—processing time increases substantially with image size. Although thinning the image prior to applying the Hough transform can improve speed, the thinning algorithm itself introduces additional overhead, rendering this approach unsuitable for real-time traffic control in isolation.

Fig 2. Character recognition from plate.

To address this limitation, the system employs a combined Hough Transform and Contour algorithm, which achieves greater accuracy and speed and is therefore suitable for real-time deployment. The overall pipeline is: Vehicle image captured → Preprocessing (Noise Removal) → Registration plate detection → Image Breakdown → PaddleOCR character recognition → Vehicle Registration Number output.

C. Image Breakdown (Segmentation)

Number plate recognition accuracy is improved through an enhanced border-detection approach. High-contrast edges that may correspond to plate boundaries are identified first. The detected edges are then transformed into a coordinate system suitable for detecting parallel lines that could form a plate boundary. Processing is restricted to regions whose aspect ratios are consistent with license plate dimensions, reducing computational load. Finally, the image is cleaned to ensure characters are clearly separated prior to recognition.

Fig 3. Indian number plate recognition.

D. Optical Character Recognition using PaddleOCR

The system employs PaddleOCR [15], an open-source OCR engine based on deep learning, which supports recognition of a wide variety of fonts and adapts to diverse lighting conditions. The detected plate image is first converted to grayscale; local contrast adjustment and image denoising are then applied to enhance text readability before the image is passed to the PaddleOCR model. The model recognizes both alphabetic and numeric characters on plates that may be deformed, of low quality, or partially obscured. In addition, a contextual verification step cross-checks the recognized character pattern against expected Indian license plate formats (e.g., state code + district code + series + number) in order to minimize recognition errors. This combination of deep learning-based OCR and pattern validation provides reliable, real-time character recognition suitable for traffic management, automated toll collection, and smart surveillance applications.

Fig 4. Shows the fetched result of number plate after detection using paddle OCR.

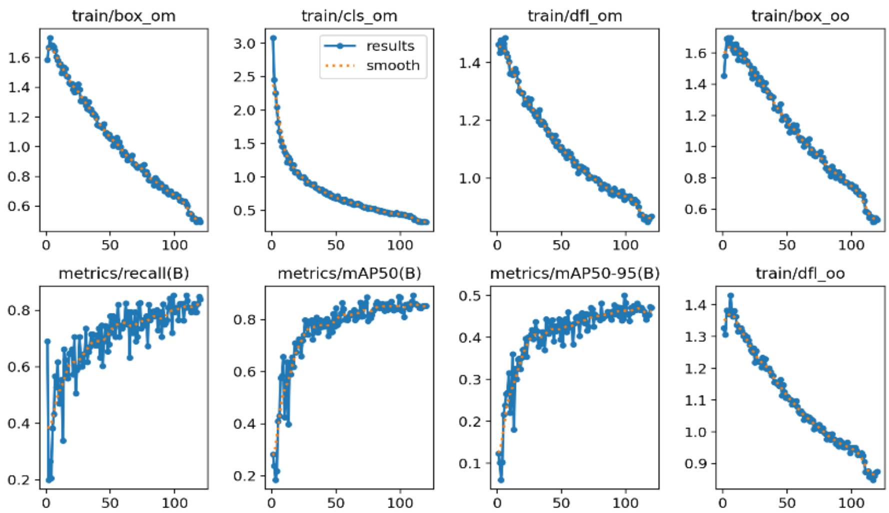

Fig 5: Shows result accuracy and smoothness.

4 Experiments

The proposed system was evaluated on two sets of Indian vehicle license plates. Images were captured manually using a physical camera across different locations and times of day to introduce variability. The implementation used Microsoft Visual C++ 14.0, executed on a Lenovo IdeaPad Gaming 3 with 8 GB RAM running Windows 11.

After registration plate detection is completed, YOLOv10 [14] is invoked for further processing. The Hough algorithm detects the plate and renders the plate candidate, after which segmentation is performed to extract individual characters. PaddleOCR [15] then recognizes the characters sequentially and renders the license plate details as output.

As shown in Figure 5, training loss curves and mean Average Precision (mAP) metrics indicate stable convergence and smooth performance. The quantitative results summarized in Figure 7 demonstrate that the model achieved a correctness rate of 99.084% (7,796 correct out of 7,868 test images), with only 72 errors and an average execution time of 0.53 seconds per image.

5 CONCLUSION

The proposed system demonstrates strong performance in recognizing vehicle license plates across varied real-world conditions, including scratched, resized, and low-resolution images. It can accurately locate and differentiate multiple plates within a single image and handles diverse vehicle types including motorcycles, cars, and trucks. These capabilities make it highly suitable for deployment in scenarios where plate visibility may be compromised.

Despite this strong overall performance, the system still encounters difficulty with severely degraded license plates, such as those that are heavily soiled or physically damaged. Continued advances in image processing and deep learning techniques are expected to address these limitations. Future work will focus on improving robustness to extreme occlusion, integrating the pipeline with IoT-enabled smart transportation infrastructure, and extending the system to support multi-language and multi-country plate formats.

Table 1: Total Results With Counted Inputs.

Column1	Column2	Column3	Column4
Model Trained Data Details			
No. Of Images		7868	
Correctness		7796	
Errors		72	
Correct %		99.084	
Error %		0.916	
Average Execution Time		0.53 S	

REFERENCES

1. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. S. Suen, "A Robust Deep Learning Approach for Automatic Iranian Vehicle License Plate Recognition for Surveillance Systems," IEEE Access, vol. 8, pp. 17–30, 2020.
2. Slimani, A. Zaarane, W. Al Okaishi, I. Atouf, and A. Hamdoun, "An automated license plate recognition system based on wavelet decomposition and CNN," Array, vol. 8, Elsevier, pp. 1–7, 2020.
3. J. Prokaj and G. Medioni, "3-D model-based vehicle recognition," in Proc. IEEE Workshop on Applications of Computer Vision (WACV), 2009, pp. 1–7, doi:10.1109/WACV.2009.5403032.
4. Y. Zhou, L. Liu, L. Shao, and M. Mellor, "DAVE: A unified framework for fast vehicle recognition and annotation," in European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 9906, 2016.
5. M. H. Zwemer, G. M. Y. E. Brouwers, R. G. J. Wijnhoven, and P. H. N. de With, "Semiautomatic training and evaluation of a learning-based vehicle make and model recognition system," Journal of Electronic Imaging, vol. 27, no. 5, 2018.
6. P. William et al., "Crime analysis using computer vision approach with machine learning," in Mobile Radio Communications and 5G Networks, Lecture Notes in Networks and Systems, vol. 588, Springer, Singapore, 2023.
7. K. B. Sathya, S. Vasuhi, and V. Vaidehi, "Perspective vehicle license plate transformation using deep neural network on genesis of CPNet," in Proc. Third International Conference on Computing and Network Communications, Procedia Computer Science, vol. 171, Elsevier, 2020.
8. F. Xie, M. Zhang, J. Zhao, J. Yang, Y. Liu, and X. Yuan, "A robust license plate recognition algorithm based on a combined feature extraction model and BPNN," Journal of Advanced Transportation, Hindawi, 2018, pp. 1–14.
9. Lubna, N. Mufti, and S. A. A. Shah, "Automatic number plate recognition: A detailed survey of relevant algorithms," Sensors, vol. 21, no. 9, p. 3028, 2021.
10. S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, pp. 311–325, 2012.
11. T. D. Duan, T. H. Du, T. V. Phuoc, and N. V. Hoang, "Building an automatic vehicle license plate recognition system," in Proc. International Conference on Computer Science RIVF, 2005, pp. 59–63.

12. J. Wei, "AlexNet: The Architecture That Challenged CNNs," Towards Data Science, 2020. [Online]. Available: https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnns-e406d5297951
13. Y. Ren and S. Lan, "Vehicle make and model recognition based on convolutional neural networks," in Proc. 7th IEEE Int. Conf. Software Engineering and Service Science (ICSESS), Aug. 2016, pp. 692–695, doi:10.1109/ICSESS.2016.7883162.
14. Wang, H. Bochkovskiy, and A. Liao, "YOLOv10: Real-Time End-to-End Object Detection," arXiv preprint arXiv:2405.14458, 2024. [Online]. Available: https://arxiv.org/abs/2405.14458
15. PaddlePaddle Authors, "PaddleOCR: Awesome Multilingual OCR Toolkits based on PaddlePaddle," GitHub Repository, 2020. [Online]. Available: https://github.com/PaddlePaddle/PaddleOCR