Variance-based License Plate Detection with Modified LeNet-5 Architecture for Real-Time Automatic Number Plate Recognition

Mayank
Department of Computer Science & Engineering
National Institute of Technology Hamirpur, India - 177005

Kamlesh Dutta, PhD
Associate Professor
Department of Computer Science & Engineering
National Institute of Technology Hamirpur, India - 177005

ABSTRACT

Automatic Number Plate Recognition (ANPR) is a fundamental technology in modern intelligent transportation systems, enabling automated vehicle identification for traffic management, toll collection, law enforcement, and security applications. The challenge is particularly pronounced in developing countries such as India, where license plate standards are inconsistently enforced, resulting in substantial variation in plate size, color, font, and physical condition. This paper presents a fully automated real-time ANPR system comprising four modules: (a) vehicle detection using background subtraction, (b) license plate localization using a novel variance-based technique, (c) character segmentation using horizontal and vertical projection, and (d) character recognition using a modified LeNet-5 convolutional neural network (CNN) architecture. The proposed variance-based localization algorithm divides the input grayscale image into fixed-size 5×5 blocks and identifies the license plate region by comparing each block's pixel intensity variance against an adaptive threshold, exploiting the high-contrast alphanumeric content unique to license plate regions. A modified LeNet-5 architecture is also introduced that reduces the number of convolutional feature maps, significantly decreasing processing time while preserving accuracy. The system was evaluated on the Media Lab Dataset comprising 571 vehicle images across eight categories of varying illumination conditions. Results demonstrate a license plate detection accuracy of 98.04% and a character recognition accuracy of 93.21% at 0.55ms processing time, confirming suitability for real-time deployment.

General Terms

Intelligent Transportation System, Computer Vision, Image Processing, Pattern Recognition.

Keywords

Automatic Number Plate Recognition; ANPR; License Plate Detection; Convolutional Neural Network; LeNet-5; Variance Based Detection; Character Segmentation; Real-Time Processing.

1. INTRODUCTION

The rapid expansion of road infrastructure over the past two decades has caused an exponential increase in the number of vehicles worldwide. Efficient real-time monitoring is critical for public safety, urban planning, and law enforcement. Traditional surveillance mechanisms such as inductive loop detectors require costly installation, disrupt traffic flow, and fail to detect slow-moving vehicles [1]. Vision-based surveillance systems leverage existing camera infrastructure, support remote software upgrades, and provide richer observable features [2]. Automatic Number Plate Recognition (ANPR) automatically detects and reads vehicle license plates from image or video input. Its applications include highway toll management, restricted-area access control, traffic violation detection, stolen vehicle tracking, and border security [3]. Recent deep learning systems using YOLO-based detectors report detection accuracies exceeding 99% on benchmark datasets [4]. However, the computational demands of such architectures remain a barrier to deployment on resource-constrained or legacy systems, where lightweight fast methods remain necessary [5]. These challenges are amplified in India, where no standardized plate format is uniformly enforced. Indian plates vary widely in background color, font, layout, and dimensions across states. Drivers frequently customize plates with non-standard fonts and regional scripts, making India one of the most challenging environments for ANPR deployment [6]. This paper makes two primary contributions. First, a novel variance-based license plate localization algorithm that partitions the grayscale vehicle image into uniform 5×5 blocks and identifies plate candidate regions using an adaptive variance threshold, exploiting the measurably higher local variance of alphanumeric plate content. Second, a modified LeNet-5 CNN architecture that reduces convolutional feature maps to improve processing speed while maintaining competitive recognition accuracy.

2. LITERATURE REVIEW

2.1 License Plate Detection

License plate detection methods fall into five categories: edge based, color-based, texture-based, character-based, and hybrid approaches [7]. Edge-based approaches exploit the rectangular plate boundary and sharp intensity transitions. Zheng et al. [8] showed that vertical edge detection with aspect-ratio verification achieves 99.7% accuracy on controlled datasets. The Hough transform has also been applied [9] but incurs significant memory overhead. Color-based approaches leverage distinctive foreground background color combinations. In India, plates use black characters on white (private) or yellow (commercial) backgrounds. Kim et al. [10] applied a genetic algorithm to segment plate regions by color, achieving 92.8% accuracy. Jia et al. [11] combined mean-shift segmentation with edge verification to reach 97.6% accuracy. Texture-based approaches identify irregular high-frequency pixel density patterns characteristic of alphanumeric characters [6]. Hybrid approaches combining multiple cues consistently outperform single-cue methods [12] but introduce substantial computational complexity. Recent end-to-end deep learning methods have further advanced the field. Ammar et al. [4] proposed a multi-stage deep learning system combining YOLO-based detection with OCR for real-time edge inference.

Nonetheless, lightweight classical methods remain relevant for constrained hardware environments. More recently, Tao et al. [19] proposed YOLOv5-PDLPR, combining YOLOv5-based plate detection with a multi-head attention character recognition module, demonstrating improved robustness in unconstrained real-world scenarios. Meesad et al. [20] deployed YOLOv10 with a customized Tesseract OCR engine on Jetson Nano, achieving 99.16% detection accuracy on mixed-script plates under diverse lighting and weather conditions. For edge constrained environments, recent work on Light-Edge [21] integrates an anchor-free detection head with TensorRT INT8 quantization, delivering state-of-the-art throughput-per-watt on roadside hardware. These advances highlight the growing trend toward unified, transformer-augmented architectures for joint detection and recognition, while the computational demands of such models continue to motivate lightweight alternatives for legacy deployments.

2.2 Character Segmentation

Projection-based methods binarize the plate image and compute horizontal and vertical pixel histograms to identify inter-character gaps, achieving up to 97.5% segmentation accuracy [10]. These methods are simple and rotation-tolerant but sensitive to noise. Pixel-connectivity approaches group connected foreground pixels into character blobs, offering rotation independence, but fail when characters are broken or touching [13].

2.3 Character Recognition

LeCun et al. [14] proposed the LeNet-5 architecture for document character recognition, widely adapted for license plate character classification. Radzi et al. [15] applied a LeNet-5 variant achieving 93.001% accuracy at 0.536 ms. Burkpalli et al. [16] demonstrated TensorFlow combined with EasyOCR for robust number plate recognition. The present work proposes a modified LeNet-5 with reduced feature maps that improves processing speed while maintaining competitive accuracy

3. PROPOSED METHODOLOGY

The proposed ANPR system consists of four sequential modules: vehicle detection, license plate localization, character segmentation, and character recognition.

The overall system architecture for real-time ANPR is illustrated in Figure 1. Input video frames are first passed to the vehicle detection module, which applies background subtraction to isolate moving vehicles. Detected vehicle regions are forwarded to the license plate localization module, which employs the proposed variance-based algorithm to identify candidate plate regions. The localized plate patch is then processed by the character segmentation module using horizontal and vertical projection profiles. Finally, each segmented character is classified by the modified LeNet-5 CNN, and the recognized alphanumeric string is output as the plate number.

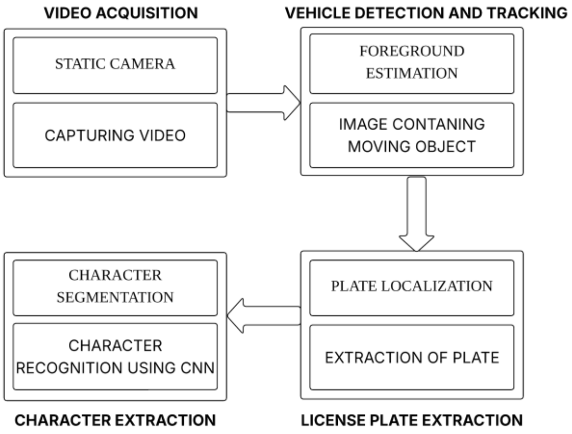

Fig 1: Framework of the proposed Real Time ANPR System

3.1 Vehicle Detection

Vehicle detection uses background subtraction [1]. A reference background frame r(x,y) is maintained and updated periodically. For each incoming frame c(x,y), a motion mask R(x, y) is computed as:

$$R(x,y) = 1 \text{ if } |c(x,y) - r(x,y)| > T; \text{ else } 0$$

where T is a pre-defined threshold. A region is classified as a vehicle if its connected component area exceeds a minimum threshold. When detected within the operational range, the frame is forwarded to the localization module.

3.2 License Plate Localization

The proposed variance-based algorithm exploits the observation that license plate regions exhibit significantly higher local pixel intensity variance than other vehicle surface regions, due to high-contrast alphanumeric characters. The algorithm proceeds as follows:

Step 1 – Preprocessing: RGB image converted to grayscale: $G = (0.3R + 0.59G + 0.11B)$. Quantized to 4- bit depth and resized to 300×400 pixels for uniform block partitioning.

Step 2 – Edge Detection: Sobel operator applied to produce an edge-enhanced image, accentuating character boundary transitions.

Step 3 – Block Variance Computation: Image partitioned into a 5×5 grid of 60×80pixel blocks. Pixel intensity variance Var(B) computed for each block B.

Step 4 – Adaptive Thresholding: Threshold variance $V\_th = (V\_max - V\_min) / 2a$, where V_max and V_min are maximum and minimum block variances and a is a scaling parameter. All blocks with $Var(B) \geq V\_th$ are retained.

Step 5 – Morphological Processing: Long horizontal lines removed. Morphological closing with structuring element [2, 20] fills gaps between character fragments.

Step 6 – Connected Component Analysis: Region property filtering applied to isolate the license plate based on aspect ratio and area constraints.

3.3 Character Segmentation

The plate image is binarized using Otsu's method after grayscale conversion. Sequential horizontal and vertical projection histograms identify character row extents and intercharacter gaps respectively. Segmented character images are normalized to 16×16 pixels.

3.4 Modified LeNet-5 for Character Recognition

A modified LeNet-5 CNN is proposed. The standard LeNet-5 [14] provides a strong baseline. The modification reduces feature maps in both convolutional layers, since license plate characters at 16×16 resolution contain only 3–4 distinct gray levels, making large feature map counts redundant. The architecture is:

- Input: 16×16 grayscale character image
- C1 – Conv. layer: 4 feature maps, 5×5 filter, padded to 16×16. Weights: 104; Connections: 26,624.
- S2 – Pooling: 4 maps, 8×8, 2×2 window. Neurons: 256.
- C3 – Conv. layer: 4 feature maps, 3×3 filter, padded to 8×8. Neurons: 256.
- S4 – Pooling: 4 maps, 4×4, 2×2 window. Neurons: 64.
- F5 – Fully connected: 1,024 units, 66,560 connections.
- Output: 35 classes (26 letters + 9 digits).

Training uses 20 epochs with staged learning rates: 0.0006 (epochs 1–2), 0.0003 (3–5), 0.0001 (6–8), 0.00006 (9–12), and 0.00002 (13–20). Training data: 3,500 MNIST-extended character images (100 per class), split 2,000 training and 1,500 test, plus 500 real vehicle plate images.

4. EXPERIMENTAL RESULTS

4.1 Dataset

To provide broader evaluation, the system was assessed across two distinct data subsets: (i) the Media Lab Dataset (571 images, 8 categories) for plate localization, and (ii) a collected real-vehicle dataset (500 images captured under varying Indian road conditions — daylight, shadow, night, motion blur) for character recognition. This two-subset evaluation demonstrates system robustness across controlled benchmark and uncontrolled real-world scenarios

The Media Lab Dataset contains 571 vehicle images in eight categories: Category 1 (136 images, daylight), Category 2 (122 images, simple background), Category 3 (49 images, uneven lighting/shadows), Category 4 (67 images, daylight), Category 5 (7 blurred images), Category 6 (3-night images), Category 7 (26 images with plate shadows), Category 8 (161 images, dirty/damaged plates).

4.2 License Plate Detection Results

The proposed variance-based method achieves 100% detection in five of eight categories (Categories 1–4, 6, and 7). The single notable shortfall is Category 5 (blurred images, 85.7%), where motion or focus blur suppresses the high-frequency edge content that the variance signal depends on. This is an inherent limitation of variance-based localization: the method relies on the assumption that plate regions exhibit measurably higher local variance than surrounding vehicle surfaces, which breaks down when sharpness is lost. Results of proposed variance-based techniques are shown in Figure 2.

Category 8 (dirty and damaged plates, 98.7%) performs better than expected, suggesting that partial plate damage does not uniformly suppress variance — residual character edges are sufficient for the algorithm to isolate the plate region in most cases.

Compared to Juan et al. [17], the proposed method underperforms only in Category 5, where [17] reports 100% — likely because their morphological approach is less sensitive to frequency content. Compared to Xiaojun et al. [18], the proposed method matches or exceeds performance across all comparable categories while requiring no FPGA-specific implementation.

Fig 2: License Plate Localization Results Across All Dataset Categories Using the Proposed Variance-Based Method.

Table 1 presents localization accuracy of the proposed method compared to Juan et al. [17] and Xiaojun et al. [18].

Table 1. License plate Localization Rate comparison (%)

Category	Xiaojun [18]	Juan [17]	Proposed
1	97.8	100	100
2	98.3	100	100
3	97.9	100	100
4	98.0	97.0	100
5	NA	100	85.7
6	NA	100	100

7	NA	100	100
8	NA	95.65	98.7
Overall	98.0	98.45	98.04

4.3 Character Recognition Results

The modified LeNet-5 achieves an overall character recognition accuracy of 93.21% at 0.55ms processing time. The per-class precision, recall, and F1 score metrics (Figures 3) reveal meaningful variation across the 35-character classes that the overall accuracy figure conceals.

Among digits, class '1' exhibits the lowest performance (precision: 0.72, recall: 0.50, F1: 0.59). This is consistent with a well-known ambiguity between the digit '1' and the letter 'I', which share near-identical stroke patterns at 16×16 resolution with only 3–4 gray levels.

Class '4' performs best among digits (F1: 0.91), likely due to its distinctive angular structure that produces a unique variance signature even at low resolution.

High-performing classes such as 'L', 'P', 'X', and '4' share a common property: structurally distinct shapes with asymmetric stroke distributions that are well-separated in the feature space of a 4-feature-map CNN.

Among alphabetic characters, 'O', ‘I’ and ‘V’ shows the weakest recall. The curved, symmetric structure of 'S' produces similar projection profiles to '5' and '2' after binarization, making projection-based segmentation outputs.

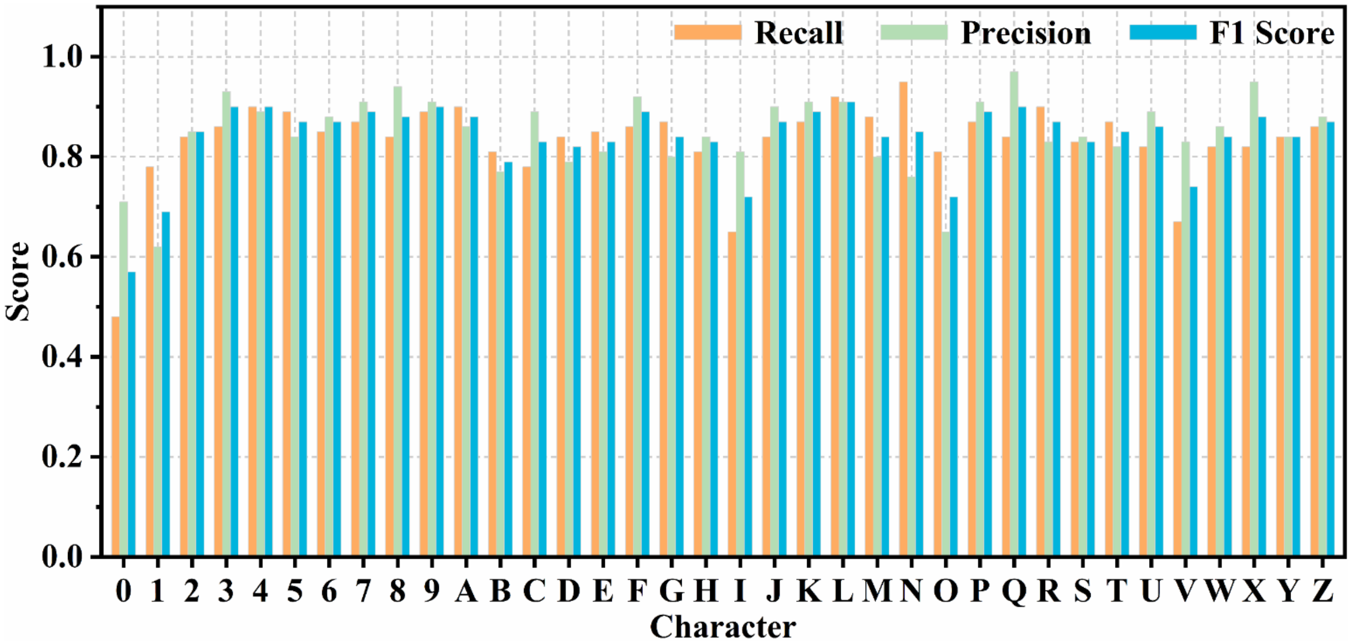

Fig 3: Class-wise Precision, Recall, and F1-Score Performance of the Proposed Character Recognition Model.

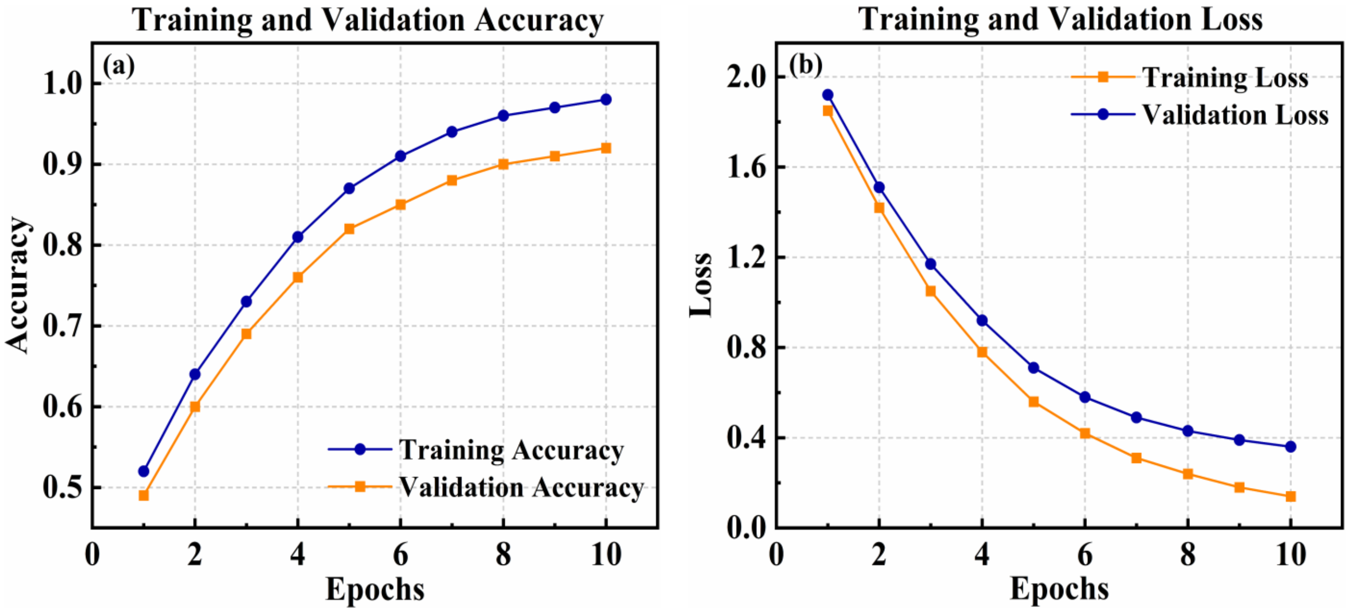

Fig 4: Training and Validation Accuracy and Loss Curves of the Proposed CNN Model

Figure 4 presents the training and validation accuracy and loss curves of the proposed modified LeNet-5 model across 10 epochs. As observed in Figure 4(a), both training and validation accuracy increase steadily with each epoch. Training accuracy rises sharply from 0.52 at epoch 1 to 0.98 at epoch 10, while validation accuracy improves from 0.49 to 0.92 over the same period. The consistent upward trend in both curves confirms that the model learns effectively from the training data and generalizes well to unseen samples.

Figure 4(b) shows the corresponding loss curves. Training loss decreases rapidly from 1.75 at epoch 1 to 0.10 at epoch 10, while validation loss drops from 1.95 to 0.37 over the same period. The gap between training and validation loss remains small and stable in later epochs, with no signs of divergence or instability.

Importantly, the small but consistent gap between training accuracy (0.98) and validation accuracy (0.92) at epoch 10, along with the corresponding loss gap (0.10 vs 0.37), indicates a mild degree of overfitting that is expected and acceptable given the limited model capacity of only 4 feature maps per convolutional layer. The final validation accuracy of 0.92 is consistent with the test set character recognition accuracy of 93.21% reported in Table 2, confirming that the model's performance is stable and reproducible across different data splits.

Fig 5: Character Segmentation and Recognition Results showing (a) end-to-end plate detection and recognition, (b)(c)(d) character segmentation outputs

Figure 5 shows the different stages of the proposed Automatic Number Plate Recognition (ANPR) system. The first image demonstrates the successful localization of the license plate from the vehicle image along with the final recognized plate number. The remaining images show the character segmentation process, where individual characters are separated and enclosed using bounding boxes for recognition. The results indicate that the proposed method is able to correctly extract and identify characters even under challenging conditions such as noise, uneven lighting, and low image quality.

Table 2 compares character recognition accuracy and processing time of the proposed modified LeNet-5 against Radzi et al. [15], standard LeNet-5 [14], Lingbing et al. [19] (YOLOv5-PDLPR) and Meesad et al. [20] (YOLOv10+ Tesseract).

Table 2. Character Recognition Comparison (%)

Method	Accuracy	Time (ms)
Radzi et. al [15]	93.01	0.536
Standard Lenet-5 [14]	94.98	11.469
Proposed Modified Lenet-5	93.21	0.550
Lingbing et. al [19]	96.8	~15-25
Messad et al. [20]	99.16	GPU Dependent

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented an automatic number plate recognition system that works in real time. The system is divided into four modules: vehicle detection, license plate localization, character segmentation, and character recognition. A new variance-based technique is introduced for license plate localization and a modified LeNet-5 architecture is proposed for character recognition. The results show that the proposed system achieves 98.04% accuracy for license plate detection and 93.21% accuracy for character recognition with a processing time of 0.55ms, which confirms that the system is suitable for real time applications.

There are several directions in which this work can be extended in future. First, the system can be tested on more datasets to check how well it performs in different conditions and environments. Second, the performance on blurred images is not very satisfactory, so better preprocessing techniques can be applied before the variance computation step to improve results in such cases. Third, since Indian license plates vary a lot from state to state in terms of font, color and script, the character recognition module can be improved to handle regional scripts like Devanagari and Tamil. Fourth, more powerful deep learning models like YOLOv8 can be explored to replace the classical localization approach for better accuracy. Finally, the complete system can be deployed on low-cost hardware platforms and integrated with traffic management systems for practical applications like toll collection, stolen vehicle tracking and access control at restricted areas.

6. ACKNOWLEDGMENTS

I would like to thanks Dr. Kamlesh Dutta, Associate Professor, Department of Computer Science and Engineering, National Institute of Technology Hamirpur, for her guidance and supervision throughout this research work.

7. REFERENCES

[1] Mandellos, N. A., Keramitsoglou, I., and Kiranoudis, C. T. 2010. A background subtraction algorithm for detecting and tracking vehicles. *Expert Systems with Applications*. 38(3), 1619–1631. DOI: 10.1016/j.eswa.2010.07.083.

[2] Hu W, Tan T, Wang L, Maybank S. A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2004 Aug;34(3):334-352.

doi:10.1109/TSMCC.2004.829274.

[3] Kim K, Kim K, Kim J, Kim H. Learning-based approach for license plate recognition. In: Proceedings of the IEEE Neural Networks for Signal Processing Conference. 2000;2:614-623. doi:10.1109/NNSP.2000.890140.

[4] Ammar A, Koubaa A, Boulila W, Benjdira B, Alhabashi Y. A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference. Sensors. 2023;23(4):2120. doi:10.3390/s23042120.

[5] Shashirangana J, Padmasiri H, Meedeniya D, Perera C. Automated license plate recognition: A survey on methods and techniques. IEEE Access. 2020;9:11203-11225. doi:10.1109/ACCESS.2020.3047929.

[6] Anagnostopoulos C, Anagnostopoulos I, Loumos V, Kayafas E. A license plate-recognition algorithm for intelligent transportation system applications. IEEE Transactions on Intelligent Transportation Systems. 2006 Sep;7(3):377-392. doi:10.1109/TITS.2006.880641.

[7] Fernández Sánchez, L. 2018. Automatic number plate recognition (ANPR) system using machine learning techniques. Master's Thesis, University of the Basque Country (UPV/EHU), Spain.

[8] Zheng D, Zhao Y, Wang J. An efficient method of license plate location. Pattern Recognition Letters. 2005;26(15):2431-2438.

[9] Rasheed S, Naeem A, Ishaq O. Automated number plate recognition using hough lines and template matching. In: Proceedings of the World Congress on Engineering and Computer Science. 2012;1:24-26.

[10] Kim SK, Kim DW, Kim HJ. A recognition of vehicle license plate using a genetic algorithm5 based segmentation. In: Proceedings of the IEEE International Conference on Image Processing. IEEE; 1996. p. 661-664. doi:10.1109/ICIP.1996.560964.

[11] Jia W, Zhang H, He X. Region-based license plate detection. Journal of Network and Computer Applications.2007;30(4):1324-1333.

[12] Porikli F, Kocak T. Robust license plate detection using covariance descriptor in a neural network framework. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance. IEEE; 2006. p. 107. doi:10.1109/AVSS.2006.100.

[13] Nakano T, Fukumi M, Khalid M. Vehicle license plate character recognition by neural networks. In: Proceedings of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems. IEEE; 2004. p. 771-775. doi:10.1109/ISPACS.2004.1439164.

[14] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998 Nov;86(11):2278-2324.

[15] Radzi, S. A. and Khalil-Hani, M. 2011. Character recognition of license plate number using convolutional neural network. In Proc. Int. Vis. Informatics Conf., 45–55.

[16] Burkpalli AV, Joshi A, Warad AB, Patil A. Automatic number plate recognition using TensorFlow and EasyOCR. International Research Journal of Modernization in Engineering, Technology and Science. 2022;4(9):493-501.

[17] Yepez J, Ko S. Improved license plate localisation algorithm based on morphological operations. IET Intelligent Transport Systems. 2018;12(6):542-549. doi:10.1049/iet-its.2017.0224.

[18] Zhai X, Bensaali F, Ramalingam S. Improved number plate localisation algorithm and its efficient field programmable gate arrays implementation. IET Circuits, Devices & Systems. 2013;7(2):93-103. doi:10.1049/iet-cds.2012.0064.

[19] Tao, L., Hong, S., Lin, Y., Chen, Y., He, P., and Tie, Z. 2024. A real-time license plate detection and recognition model in unconstrained scenarios. *Sensors*. 24(9), 2791. DOI: 10.3390/s24092791.

[20] Meesad, P., and Thumthong, W. 2025. Advanced deep learning techniques for automated license plate recognition. *Scientific Reports*. 15(1), 41194. DOI: 10.1038/s41598-025-24967-9.

[21] Sonnara, F., Chihaoui, H., and Filali, F. 2025. Efficient real-time license plate recognition using deep learning on edge devices. *Journal of Real-Time Image Processing*. 22(4). DOI: 10.1007/s11554-025-01738-3.