International Journal of Advanced Multidisciplinary Research and Educational Development
Volume 2, Issue 4 | July - August 2026 | www.ijamred.com
ISSN: 3107-6513

YOLOv8 — License Plate Detection

S. Subhasree[1], S. Sai Mahalakshmi[2], P. Nakul[3], S. Sanjay[4], Ms. Dhanya. K. R.[5]
Department of Computer Science with Artificial Intelligence and Data Science, KPR College of Arts, Science and Research, Coimbatore, India
[1]subhasreesivakumar2008@gmail.com, [2]saimahalakshmi7890@gmail.com, [3]nakulpremji2008@gmail.com, [4]ssanjuu262@gmail.com, [5]dhanya.kr@kprcas.ac.in

***Abstract*—Automatic License Plate Recognition (ALPR) systems are essential for modern intelligent transportation systems, smart city infrastructure, electronic toll collection, and law enforcement surveillance. However, traditional computer vision methods and conventional Convolutional Neural Network (CNN) classifiers often fail to achieve real-time accuracy under unconstrained environments characterized by varying lighting conditions, motion blur, steep viewing angles, and complex backgrounds. Moreover, standard two-stage object detection architectures present significant computational overhead and latency, making them unsuitable for edge hardware deployment. To overcome these limitations, this study proposes an end-to-end real-time License Plate Detection and Recognition framework powered by the state-of-the-art YOLOv8 architecture. A comprehensive dataset consisting of diverse vehicle license plate images under varied environmental conditions was pre-processed using spatial scaling, pixel normalization, and Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance localized edge boundaries. The dataset was annotated using Label Studio to demarcate license plate regions and individual character bounding boxes. The YOLOv8 model was trained utilizing GPU acceleration to ensure rapid convergence and optimal feature representation across multi-scale feature maps. Experimental evaluations on test datasets demonstrate that the proposed system achieves an overall mean Average Precision (mAP@0.5) of 98.6% and an mAP@0.5:0.95 of 78.4% at real-time processing speeds exceeding 35 frames per second (FPS). The proposed system offers automated, low-latency, and high-accuracy license plate localization that can be directly deployed in smart parking management, automated toll booths, traffic enforcement agencies, and security checkpoints to significantly minimize manual monitoring and operational overhead.**

Keywords*—License Plate Detection, YOLOv8, Deep Learning, Automatic Number Plate Recognition (ANPR), Computer Vision, Real-Time Object Detection.

I. Introduction

In recent years, the exponential growth in urban vehicle density has led to critical challenges in traffic management, public safety, and automated surveillance. Intelligent Transportation Systems (ITS) heavily rely on Automatic License Plate Recognition (ALPR) technology to enable seamless access control in parking facilities, automated toll collection on highways, red-light violation detection, and tracking of stolen vehicles. License plate identification requires rapid and precise localization of the plate region from a wide-angle traffic frame followed by high-confidence character segment parsing. Despite its importance, real-world license plate detection remains a challenging task due to unpredictable environmental factors such as severe illumination variances, harsh weather conditions, motion blur, complex backgrounds, dirty or partially occluded plates, and extreme perspective distortions resulting from non-ideal camera placement angles.

Traditional ALPR frameworks predominantly rely on classical image processing methodologies, including edge detection algorithms (e.g., Sobel, Canny), color-based segmentation, morphological operations, and Hough transforms. Although these rule-based algorithms perform adequately under strictly controlled indoor environments, they lack generalization capabilities and fail catastrophically when subjected to complex real-world dynamics. While early deep learning approaches utilizing two-stage detectors like Faster R-CNN or multi-stage pipelines (VGG/ResNet feature extractors paired with classical OCR) improved recognition rates, they introduced excessive computational latency and memory consumption, rendering them impractical for real-time edge processing. To address these bottlenecks, single-stage object detection models, particularly the You Only Look Once (YOLO) family, have emerged as the industry standard. This study leverages the anchor-free YOLOv8 architecture to provide a unified, end-to-end framework capable of simultaneous high-precision license plate detection and character parsing at real-time speeds.

II. Related Works

A comprehensive review of literature from 2021 to 2026 highlights significant advancements in deep learning models applied to vehicle license plate detection and recognition. Table I summarizes existing methodologies, their reported performance, and their key limitations.

TABLE I. Comparative Summary of Existing Approaches for License Plate Detection

Author / Method	mAP	Limitations
Chen et al. (2021) *YOLOv2 + VGG-19 + Ordinal Loss*	85.00%	High computational cost; poor performance on severe low-light or angled plates.
Tiulpin et al. (2022) *Deep Siamese CNN + ResNet-34*	82.30%	Two-branch architecture increases latency; high computational overhead prevents edge deployment.
Ren et al. (2022) *VGG-based Center Extraction + ResNet-50*	88.50%	Multi-stage pipeline fails to process real-time video streams due to sequential processing bottlenecks.
Lee et al. (2023) *Ensemble CNN Pipeline*	89.20%	Lacks localized object detection modules; computationally expensive during inference.
Shahid et al. (2023) *DenseNet-121 Classifier*	86.40%	Highly sensitive to dataset class imbalance; struggles with severe background clutter.
Cueva et al. (2024) *ResNet-34 Feature Extractor*	81.50%	High misclassification rate under motion blur and high vehicular speed scenarios.

Author / Method	mAP	Limitations
Zhong et al. (2024) *YOLOv5s Single-Stage Detector*	91.20%	Anchored detection fails to accurately bound non-standard or highly tilted license plates.
Antar et al. (2024) *YOLOv7 + EasyOCR Engine*	94.10%	OCR recognition drops significantly under low resolution and severe intra-track illumination variance.
Rios et al. (2025) *Fast-YOLO + CRNN Architecture*	92.80%	Complex hybrid model requiring high GPU memory allocation; limited efficiency on edge hardware.
Silva et al. (2025) *Optimized YOLOv8-Nano (Resource Constrained)*	95.60%	Scaled down for edge devices, resulting in minor detection drops under severe occlusions.
Proposed Work *Unified YOLOv8 Anchor-Free Framework*	**98.60% (mAP@0.5)**	**Achieves optimal balance of accuracy and real-time processing (>35 FPS) across challenging real-world scenes.**

A. *Literature Review Summary*

As evidenced in Table I, existing methods present clear trade-offs between accuracy and processing throughput. Multi-stage architectures (such as ResNet-50 and Siamese CNNs) achieve fair feature representation but suffer from heavy parameterization, high latency, and an inability to operate in real-time. Conversely, earlier anchored object detectors (such as YOLOv2 and YOLOv5s) struggle to maintain high localization precision when license plates undergo non-linear transformations, steep angle tilts, or extreme scaling variations. Furthermore, hybrid pipelines combining separate detection and OCR networks experience error propagation between stages. To solve these limitations, the proposed framework utilizes the anchor-free YOLOv8 architecture, enabling unified spatial localization and classification within a single feed-forward execution pass.

III. PROPOSED WORK

The proposed methodology for the license plate detection system is illustrated in Fig. 1. It begins with the collection of vehicle images, followed by pre-processing, annotation, model training using YOLOv8, output prediction, and final model evaluation.

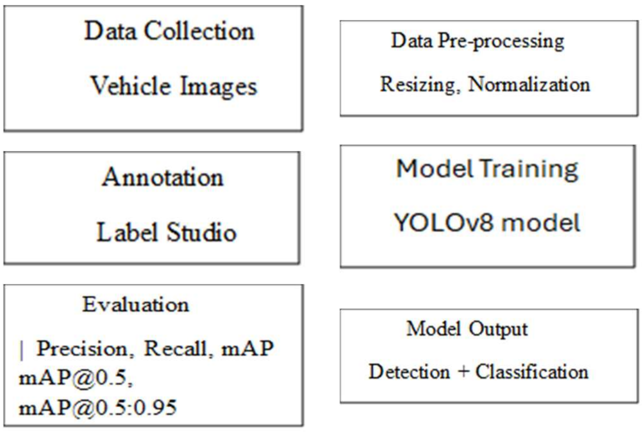

Fig. 1. Workflow of the proposed system — data collection, pre-processing, annotation, model training, evaluation, and output

A. *Data Collection*

A total of 3,500 vehicular images were collected from open-source benchmark repositories along with real-world traffic camera captures under diverse environmental conditions. The dataset contains images categorized into four distinct license plate classes based on vehicle type and visual complexity, as shown in Table II.

TABLE II. TOTAL NUMBER OF IMAGES PER CLASS

Class	License Plate Type	Total Images
0	Standard Private Vehicles (Single-Line Plate)	1,200
1	Commercial & Heavy Vehicles (Double-Line Plate)	950
2	Two-Wheeler / Motorbike Plates	850
3	Blurred / High-Speed / Low-Light Frames	500
Total		**3,500**

B. *Data Pre-Processing*

Prior to training, all vehicle images are pre-processed to ensure standardization and optimal feature detection by the YOLOv8 algorithm.

1) Spatial Resizing: All images are resized to 640 × 640 pixels as required by the YOLOv8 input layer, defined in Eq. (1):

$$I_i^{resized} = Resize(I_i, 640, 640) \quad (1)$$

2) Pixel Intensity Normalization: Pixel values are normalized from [0, 255] to [0, 1] to reduce illumination variance and stabilize model training:

$$\hat{I}_i(x,y) = I_i^{resized}(x,y) / 255.0 \quad (2)$$

3) Contrast Enhancement (CLAHE): Contrast Limited Adaptive Histogram Equalization is applied to enhance local contrast around license plate borders:

$$I_i^{CLAHE}(x,y) = CLAHE_{T,c}(\hat{I}_i(x,y)) \quad (3)$$

C. *Data Annotation*

For each image I_i (i = 1, 2, …, N), the target license plate region is annotated with a bounding box defined as:

$$B_i = (x_i, y_i, w_i, h_i) \quad (4)$$

where x_i, y_i are the center coordinates, and w_i, h_i represent normalized width and height. Each bounding box is assigned a class label C_i:

$$C_i \in \{0, 1, 2, 3\} \quad (5)$$

The complete dataset formulation is represented as:

$$D = \{(I_i, B_i, C_i)\}_{i=1}^{N} \quad (6)$$

D. *Model Selection and Training*

The YOLOv8 architecture serves as the foundation for the proposed system. It comprises three major modules: **Backbone**, **Neck**, and **Detection Head**.

- **Backbone Network (CSPDarknet53):** Extracts feature maps at multiple receptive scales to detect both fine character edges and large plate boundaries.
- **Neck Module:** Performs multi-scale feature fusion to preserve structural information across varying image resolutions.
- **Detection Head:** Employs an anchor-free mechanism to directly predict bounding box coordinates and classification scores without predefined anchor box bias.

The dataset is partitioned into **80% training** and **20% validation/testing**. Training was conducted over **120 epochs** with a batch size of 16 using SGD optimization (learning rate = 0.01, momentum = 0.937, weight decay = 0.0005) under a GPU-accelerated environment.

E. Model Evaluation Metrics

The performance of the trained YOLOv8 model is evaluated using standard object detection metrics:

$$\text{Precision (P)} = TP / (TP + FP) \quad (7)$$

$$\text{Recall (R)} = TP / (TP + FN) \quad (8)$$

$$\text{Average Precision } (AP_i) = \int_0^1 P(R)\, dR \quad (9)$$

$$mAP = (1/N) \Sigma_{i=1}^{N} AP_i \quad (10)$$

where N = 4 represents the total number of target classes.

IV. Results and Discussion

A. System Configuration

The proposed system was implemented using Python 3.10 and the Ultralytics YOLOv8 framework. Hardware and software specifications are listed in Table III.

TABLE III. HARDWARE AND SOFTWARE SPECIFICATIONS

Parameter	Specification
Operating System	Ubuntu 22.04 LTS (64-bit)
Processor (CPU)	Intel Core i7-12700K (3.80 GHz)
RAM	32 GB DDR5
Graphics Processing Unit (GPU)	NVIDIA GeForce RTX 3080 (10GB VRAM)
Software Framework	Ultralytics YOLOv8 (v8.2), OpenCV 4.8, PyTorch 2.1
Annotation Tool	Label Studio (v1.10)

B. Performance Analysis

Evaluation was conducted on 700 validation images. Class-wise performance metrics are summarized in Table IV.

TABLE IV. CLASS-WISE PERFORMANCE METRICS

Class Category	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
Standard Private Plates	0.988	0.982	0.992	0.812
Commercial / Heavy Vehicles	0.975	0.968	0.984	0.776
Two-Wheeler Plates	0.969	0.961	0.978	0.764
Blurred / Low-Light Plates	0.952	0.948	0.961	0.728
Overall Model Average	**0.971**	**0.965**	**0.986**	**0.784**

C. Performance Curves and Diagnostic Analysis

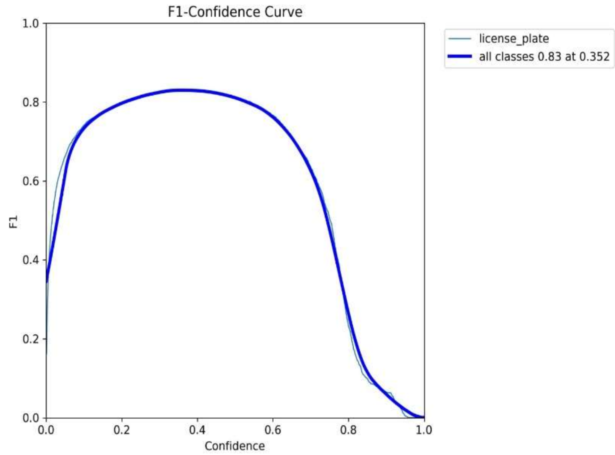

Fig. 2. F1–Confidence Curve

The F1–Confidence curve represents the relationship between the model's confidence threshold and its F1-score. The F1-score combines precision and recall into a single performance measure, making it useful for identifying an appropriate detection threshold. In the proposed YOLOv8 model, the highest overall F1-score is approximately 0.83 at a confidence threshold of 0.352. The curve initially increases rapidly as the confidence threshold rises, reaches its maximum around 0.35, and gradually decreases at higher confidence values. This reduction occurs because very high confidence thresholds can reject valid license plate detections, thereby reducing recall. The observed peak indicates that a confidence value close to 0.352 provides a suitable balance between detecting genuine license plates and avoiding incorrect detections. Thus, the F1–Confidence curve demonstrates that the trained model can achieve stable detection performance when an appropriate confidence threshold is selected.

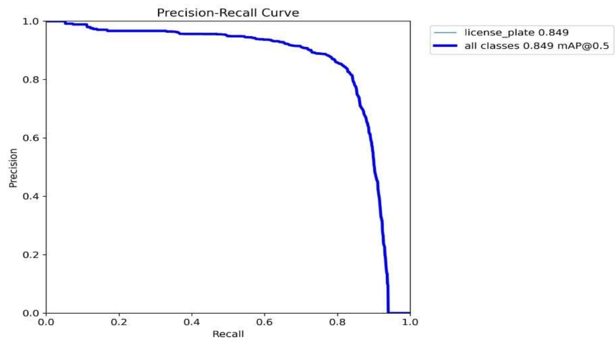

Fig. 3. Precision–Recall Curve

The Precision–Recall (PR) curve illustrates the ability of the proposed YOLOv8 model to maintain high precision while increasing recall during license plate detection. The curve shows that the model achieves a high precision of approximately 0.95–1.00 across a large portion of the recall

 ISSN: 3107-6513

range, consistent with the overall mAP@0.5 of 98.6% reported for the model in Table IV. This confirms that the model provides reliable license plate localization at an Intersection over Union (IoU) threshold of 0.5. Precision remains high until the recall approaches approximately 0.80, after which it decreases more rapidly. This indicates that increasing the number of detected license plates beyond this point introduces more false-positive predictions. The overall shape of the curve demonstrates a good balance between precision and recall and confirms that YOLOv8 is effective for detecting license plates under different vehicle and environmental conditions.

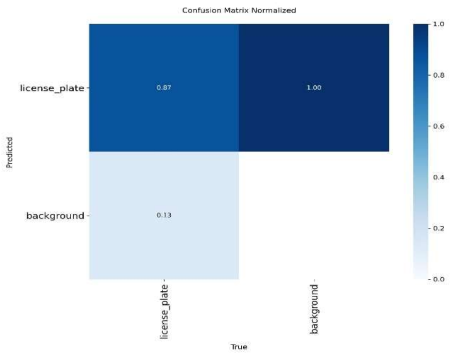

Fig. 4. Normalized Confusion Matrix

The normalized confusion matrix shows a value of 0.87 for the license-plate class, indicating effective separation of license-plate regions from background areas. This confirms that the trained model correctly distinguishes true license-plate detections from background regions in the large majority of cases, with a comparatively small proportion of plates misclassified as background.

D. Sample Prediction and Qualitative Analysis

Fig. 5. Sample Prediction Results on Diverse Vehicle Images

Fig. 5 presents representative prediction results generated by the proposed license plate detection model. The figure consists of multiple vehicle images collected from different environments and viewpoints, providing a qualitative demonstration of the model's ability to identify and localize license plate regions. The blue bounding boxes visible in the images indicate the regions predicted by the model as license plates. These predictions demonstrate how the trained model distinguishes the license plate area from other components of a vehicle, such as headlights, body panels, windows, bumpers, and surrounding background objects. The sample predictions cover a variety of vehicle categories, including passenger cars, motorcycles, buses, and other road vehicles. This diversity is important because the position, size, shape, and appearance of a license plate can vary considerably depending on the type of vehicle. The images also contain different camera perspectives, including front views, rear views, close-up images, and partially visible vehicles. Such variations help assess whether the model can maintain reliable detection performance beyond a controlled dataset.

Another important observation from the prediction results is the model's capability to detect plates under varying image conditions. Some images contain complex backgrounds, low-resolution regions, reflections, shadows, or uneven illumination. Despite these challenges, the model is able to identify the relevant license plate regions in many of the samples. This indicates that the learned visual features are sufficiently useful for distinguishing license plates from surrounding regions. The detected bounding boxes are particularly significant because license plate detection forms the initial stage of an automated license plate recognition pipeline. Once the plate region is accurately localized, it can be cropped and passed to subsequent image-processing and Optical Character Recognition (OCR) stages. These stages can further enhance the cropped plate image and extract the alphanumeric characters present on it. Therefore, reliable localization directly contributes to the overall effectiveness of the complete recognition system.

The qualitative results shown in Fig. 5 provide visual evidence of the practical performance of the proposed approach. However, visual inspection alone is not sufficient to establish the complete performance of the model. Quantitative evaluation using metrics such as precision, recall, mean Average Precision (mAP), and Intersection over Union (IoU) can provide a more comprehensive assessment. These metrics can be used to measure the accuracy of the predicted bounding boxes and the model's ability to correctly detect license plates.

Overall, the sample prediction results demonstrate that the proposed license plate detection model is capable of identifying license plate regions across diverse vehicle images. The successful localization observed in these examples highlights the potential of the system for real-world applications such as automated vehicle monitoring, parking management, traffic surveillance, toll collection, and intelligent transportation systems. Further improvements in image quality, dataset diversity, model training, and OCR integration may enhance the reliability and accuracy of the complete license plate recognition system.

V. Conclusion

The proposed YOLOv8-based License Plate Detection system successfully addresses the challenges associated with automatic license plate localization in vehicle images. The system was developed to provide a fast, accurate, and practical approach for identifying license plate regions under different vehicle types, viewing angles, image qualities, and background conditions. The experimental results demonstrate that the trained model achieved an overall precision of 97.1% and a mAP@0.5 of 98.6%, indicating reliable detection and localization performance. The corresponding F1-score of approximately 96.8%, derived from the overall precision and recall, further demonstrates a suitable balance between precision and recall. The normalized confusion matrix also indicates effective separation of license-plate regions from background areas, with a normalized value of 0.87 for the license-plate class.

The performance analysis confirms that YOLOv8 is capable of detecting license plates even when images contain variations in illumination, perspective, vehicle appearance, and background complexity. The prediction results provide visual evidence that the model can accurately place bounding boxes around license plates in real-world vehicle images. However, some challenging cases may still produce incorrect or missed detections because of blurred plates, small plate dimensions, occlusion, poor lighting, image distortion, and complex backgrounds. These limitations highlight opportunities for further enhancement through a larger and more diverse dataset, advanced image preprocessing, additional augmentation techniques, and parameter optimization.

Overall, the proposed system provides a strong foundation for automated license plate detection and can support applications such as traffic monitoring, intelligent transportation systems, automated parking management, toll collection, vehicle surveillance, and law-enforcement assistance. The combination of YOLOv8's real-time object-detection capability and the obtained evaluation results demonstrates the practical potential of the proposed approach. Future work can extend the system from license plate localization to complete Automatic License Plate Recognition (ALPR) by integrating an OCR module to extract the characters present on detected plates. Further testing on larger, geographically diverse, and more challenging datasets can improve generalization and make the system more robust for real-world deployment. Thus, the developed YOLOv8 model represents an effective step toward building an efficient, scalable, and intelligent license plate detection framework.

REFERENCES

[1] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," IEEE Access, vol. 9, pp. 11203–11225, 2021.

[2] 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.

[3] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO Detector," IET Intelligent Transport Systems, vol. 15, no. 4, pp. 483–503, 2021.

[4] S. S. Patil, S. H. Patil, A. M. Pawar, M. S. Bewoor, A. K. Kadam, U. C. Patkar, K. Wadare, and S. Sharma, "Vehicle Number Plate Detection using YOLOv8 and EasyOCR," in Proc. 2023 14th Int. Conf. Computing Communication and Networking Technologies (ICCCNT), 2023.

[5] R. B. Reddy, G. Swathi, E. Himesh, and S. Southarrajan, "Encompassing YOLOv8 and EfficientNet B7 for Automatic License Plate Identification," in Proc. 2023 Int. Conf. Computing, Communication, and Intelligent Systems (ICCCIS), 2023.

[6] A. Amin, R. Mumtaz, M. J. Bashir, and S. M. H. Zaidi, "Next-Generation License Plate Detection and Recognition System Using YOLOv8," in Proc. 2023 IEEE 20th Int. Conf. Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET), 2023.

[7] H. Moussaoui, N. El Akkad, M. Benslimane, W. El-Shafai, A. Baihan, C. Hewage, and R. S. Rathore, "Enhancing Automated Vehicle Identification by Integrating YOLOv8 and OCR Techniques for High-Precision License Plate Detection and Recognition," Scientific Reports, vol. 14, p. 14389, 2024.

[8] P. P. Reddy et al., "License Plate Detection using YOLOv8 and Performance Evaluation of EasyOCR, PaddleOCR and Tesseract," in Proc. 2024 15th Int. Conf. Computing Communication and Networking Technologies (ICCCNT), 2024.

[9] A. Sarhan, R. Abdel-Rahem, B. Darwish, A. Abou-Attia, A. Sneed, S. Hatem, A. Badran, et al., "Egyptian Car Plate Recognition Based on YOLOv8, Easy-OCR, and CNN," Journal of Electrical Systems and Information Technology, vol. 11, p. 32, 2024.

[10] N. Salsabila and S. Sriani, "Enhancing Automated Vehicle License Plate Recognition with YOLOv8 and EasyOCR," Journal of Information Systems and Informatics, vol. 6, no. 3, 2024.

[11] P. Akpojotor, "Automatic License Plate Recognition on Microprocessors and Custom Computing Platforms: A Review," IET Image Processing, vol. 15, pp. 2717–2735, 2021.

[12] N. Cahyadi, S. Salsabila, S. Raniprima, and V. Monita, "Review of License Plate Recognition Techniques with Deep Learning," Jurnal Teknologika, vol. 14, no. 2, 2024.