Automatic Number Plate Recognition using YOLO v8 and OCR : A Review

Bhavya Shah*, Arohi Patel**
*(Computer Science & Engineering, Dr. Kiran & Pallavi Patel Global University, Varnama,Gujarat,India
Email: 28bhavyashah@gmail.com
** (Computer Science & Engineering, Dr. Kiran & Pallavi Patel Global University, Varnama,Gujarat,India
Email: arohipatel3010@gmail.com)

--************************----------------------------------

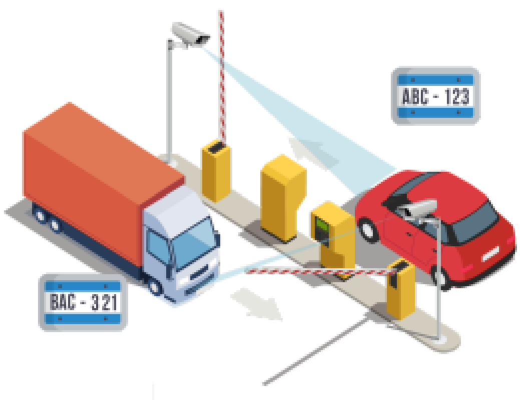
Abstract:

Automatic Number Plate Recognition (ANPR) has become a vital technology for intelligent transportation, traffic monitoring, and security applications. Traditional image-processing methods often face challenges under real-world conditions due to variations in lighting, weather, camera quality, and the diversity of license plate designs. This study presents an AI-driven ANPR system using YOLO v8 for precise license plate detection and advanced OCR techniques for accurate character recognition. The proposed framework involves comprehensive dataset preparation, preprocessing, YOLO v8-based plate detection, extraction, and recognition using OCR engines such as PaddleOCR and LPRNet. Additional enhancements, including image augmentation, noise reduction, and contrast optimization, improve performance under complex conditions. Comparative analysis shows that this approach outperforms traditional methods in accuracy, speed, and robustness, making it suitable for real-time deployment. The system has significant potential for applications such as toll collection, smart parking, traffic enforcement, and surveillance. Future work may include multilingual OCR, explainable AI, and lightweight models for wider practical adoption.

Keywords — Automatic Number Plate Recognition (ANPR); YOLO; Deep Learning; Object Detection; Optical Character Recognition (OCR); PaddleOCR; License Plate Detection; Smart Transportation; Computer Vision; Image Processing; Intelligent Traffic Monitoring; Real-Time Recognition; Preprocessing; Character Segmentation; Edge Deployment.

--************************----------------------------------

I. INTRODUCTION



Automatic Number Plate Recognition (ANPR) has become an important technology in transportation, security surveillance, automated tolling, traffic management, and law enforcement. Recent studies highlight the rising demand for accurate and real-time ANPR solutions as vehicle numbers and automation needs grow worldwide [1].

ANPR systems usually include detecting the number plate, segmenting the characters, and recognizing the alphanumeric text. This process becomes significantly difficult in real-world conditions due to changes in lighting, weather, camera quality, and plate design [5]. Since number plates vary in font, color, size, language, and layout across different areas, creating a general and flexible recognition model is a complex challenge [8].

Traditional ANPR methods relied on image-processing techniques like edge detection, thresholding, and morphological operations. While these techniques worked well in controlled settings, their performance dropped noticeably when plates were blurred, tilted, obscured, or captured in poor lighting [2]. To address these issues, recent research has increasingly turned to deep-learning techniques, particularly object detectors like YOLO and OCR frameworks, which provide better reliability in various conditions [9].

Newer models such as YOLOv8, YOLOv10, and YOLOv12 have demonstrated significant gains in plate detection accuracy. Meanwhile, OCR systems like PaddleOCR, Tesseract, and LPRNet improve character recognition across different plate formats [3]. Many studies also show that preprocessing techniques like image enhancement, de-noising, and perspective correction can boost recognition performance in tough scenarios [4]

Despite these advances, several important research gaps still exist. Many models are trained using small, 1 region-specific datasets, which limits their generalization across countries [10]. Some models use complex neural networks not suited for real-time operation on laptops or edge devices, limiting their practical use [11]. Environmental factors like rain, fog, night glare, and motion blur continue to hinder detection and OCR accuracy [7].

Another significant gap is the lack of explainable AI (XAI). Most deep-learning models function as "black boxes," which undermines trust and understanding in real-world applications [13]. In addition, dataset imbalances, missing values, and insufficient representation of rare plate styles further lead to inconsistent results [9][15].

Given these challenges, this research aims to create an AI-powered ANPR system that combines advanced deep-learning detectors, strong preprocessing, better data-balancing strategies, effective feature-selection methods, and components for explainable AI. This approach aims to develop a more accurate, reliable, and understandable ANPR model that works well in diverse and challenging real-world situations [10].

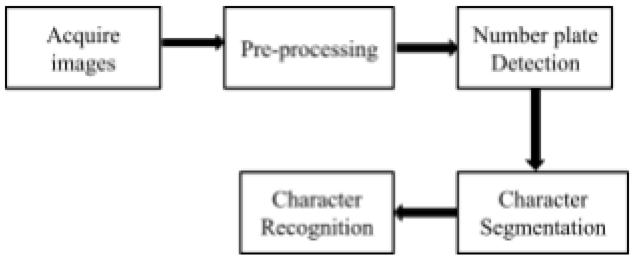


Fig 1.1 Block diagram of Number Plate Recognition System [8]

II. LITERATURE REVIEW

A number plate detection technique called YOLOv8s was utilized in this research. The technique was incorporated with preprocessing and a distributed edge server architecture. The method worked well under different situations but failed when the images were heavily degraded and complex noise was involved. Generalization was restricted to Qatar number plates only. Attempts to enhance deep learning accuracy through classical preprocessing were unsuccessful [1].

This paper applied the latest YOLOv12 for detection and PaddleOCR for character recognition, focusing on Romanian plates. The method improved speed and accuracy over previous YOLO versions. However, the dataset was relatively small and region-specific, limiting cross-country performance. OCR struggled under low-light conditions, and no edge-hardware deployment tests were included. The model lacked robustness evaluation for extreme environments [2].

This work used YOLOv10 for plate detection and a customized Tesseract OCR for dual Thai–Roman scripts. It achieved strong accuracy on a large dataset, outperforming several older YOLO versions. However, the approach was limited to Thai plates, reducing global generalization. OCR still showed errors for stylized fonts, and deep-learning performance depended heavily on clean images. Only Jetson Nano was tested, restricting hardware diversity [3].

This study combined YOLOv8 with EasyOCR and CNN classification to recognize Egyptian plates. Performance was strong for multi-script Arabic–English characters and complex plate backgrounds. Still, the dataset was moderately sized and lacked night-time and weather variations. OCR accuracy declined under blur and shadow. No comparison with newer detectors like YOLOv10 or YOLOv12 was included [4].

This paper introduced a compact "Light-Edge" model combining detection and recognition for real-time ANPR on low-power devices. It achieved fast inference and good robustness under blur and glare. However, the system relied on limited dataset variations and lacked multilingual plate experiments. No comparison with modern OCR frameworks was performed. Performance in extreme weather was not fully explored [5].

This study proposed a lightweight ANPR system combining an optimized YOLOv8n with an improved LPRNet for character recognition. The model reduced detection parameters by 32% while maintaining accuracy, making it suitable for edge devices. Enhancements like BiFPN and GCE modules improved small-plate detection. However, experiments were dataset-specific, and robustness under extreme conditions like rain or night glare was not deeply evaluated [6].

This paper introduced "Light-Edge," an end-to-end ALPR model integrating ResNet18-FPN with an anchor-free head and CTC decoder. It achieved 14 FPS on Jetson Nano while consuming only 4.8 W, outperforming models like AF-Net and YOLO-MobileLPR. Strengths include high speed and low power usage for real-time deployment. However, the system was tested only on CCPD-based datasets, and global multi-script plate validation was limited [7].

This survey explored ANPR methods and proposed integrating YOLOv8 with PaddleOCR for improved 2 multilingual character recognition. It highlighted the weaknesses of EasyOCR and classical pipelines under skewed, low-light, and multilingual conditions. While the survey identified strong potential for deep-learning-based OCR, it did not introduce a new model or experimental validation. Lack of comparative performance metrics remains a limitation [8].

This paper presented a hybrid framework focusing on plate localization with a lightweight CNN and character recognition through a compact OCR module. It improved inference time and resource efficiency for embedded devices. The model performed well on clean datasets but showed accuracy drops under blur, weather disturbances, and angled plates. The absence of multilingual evaluation and advanced detectors like YOLOv8/YOLOv10 limited its scope [9].

This work applied a multi-stage deep-learning pipeline using CNN-based detection followed by segmentation-free OCR. The method improved recognition reliability compared to classical segmentation approaches. However, the system required high-quality images and struggled with night glare, occlusion, and motion blur. The lack of end-to-end integration and no optimization for edge devices reduce real-time applicability [10].

This study proposed an ANPR pipeline customized for Indian traffic using YOLOv5 for vehicle detection, StrongSORT for tracking, and a transfer-learned ALPR model for oblique plate detection. Majority voting with Levenshtein distance improved OCR accuracy across multiple frames. The system performed well on real-world Indian traffic videos but showed reduced OCR accuracy for low-quality and fast-moving inputs. It is computationally heavy on CPUs, limited to Indian plates, and lacks multilingual support and explainability [11].

This paper introduced a real-time license plate detection framework using YOLO with LSTM-based OCR, achieving high accuracy and fast processing compared to traditional methods. Evaluation of multiple YOLO variants showed a clear trade-off between speed and precision. The approach primarily focused on plate detection rather than robust multilingual recognition. Limitations include restricted dataset diversity, lack of edge-device optimization, and no analysis of extreme weather, night-time conditions, or explainability [12].

This paper presents a real-time Automatic Number Plate Recognition (ANPR) system using YOLOv5 for license plate detection and Tesseract OCR for character recognition. The system performs well in challenging conditions such as low lighting and different viewing angles. The processing pipeline includes image preprocessing, plate detection, extraction, segmentation, and OCR-based recognition. However, the approach depends on traditional OCR, lacks multilingual support, and does not use advanced deep-learning sequence models. The study also has limitations in dataset diversity and does not address explainability (XAI) or optimization for edge devices [13].

This study developed a YOLOv8-based ANPR system for Qatari license plates, integrating enhanced datasets and preprocessing techniques like edge detection and k-means thresholding. It achieved strong recognition accuracy and provided a practical, deployment-ready framework with API support. However, the model was tested mainly on local datasets and lacks explainability or cross-country validation [14].

This paper introduces a real-time ANPR system using YOLOv11 for license plate detection and PaddleOCR for text recognition, achieving up to 98% accuracy. Techniques like CLAHE preprocessing improve performance in low-light and noisy conditions, while a MySQL database enables automated record storage. However, the system lacks cross-country and multilingual evaluation, does not include explainability (XAI), requires high-end hardware, and is not tested on large-scale real-world traffic datasets [15].

III. Proposed Methodology

The aim of methodology for building an Automatic Number Plate Recognition (ANPR) system using YOLO consists of several sequential steps, starting from dataset preparation to plate detection, text recognition, and final output generation. The complete workflow is described below:

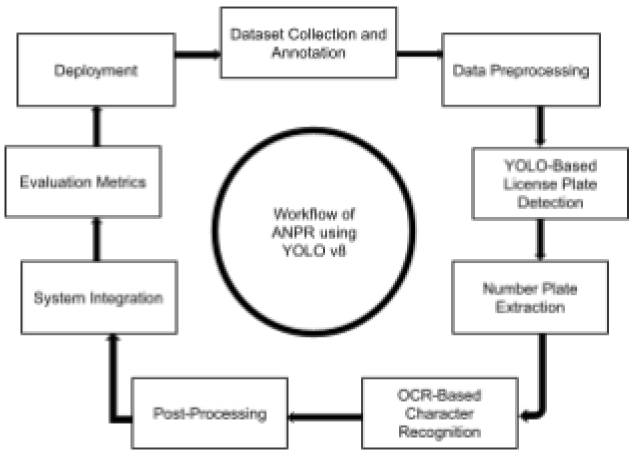


Fig 3.1 Workflow of ANPR using YOLO v8 [13]

1. Dataset Collection and Annotation: Collect images of vehicles from CCTV feeds, traffic cameras, parking lots, or open-source datasets. Ensure dataset variety: different regions, lighting conditions (day/night), angles, blur levels, plate

colors, fonts, and backgrounds. Annotate license plates using tools such as LabelImg, Roboflow, CVAT, creating bounding boxes around plates. Export annotations in YOLO format (.txt files with class and coordinates).

2.Data Preprocessing: Resize images to YOLO-supported resolutions (e.g., 640×640). Apply augmentation: rotation, blur, brightness adjustments, noise insertion, occlusion simulation, and weather simulation (rain/fog).Normalize images to improve model generalization. Split dataset into train (70%), validation (20%), and test (10%).

3.YOLO-Based License: Plate Detection Choose a YOLO model (YOLOv8, YOLOv9, YOLOv10, YOLOv11, or YOLOv12) depending on speed and accuracy requirements. Train YOLO using annotated plate images with parameters such as:

- Batch size
- Learning rate
- Optimizer (Adam/SGD)
- Number of epochs

Monitor training using metrics like mAP50, mAP50-95, Precision, Recall, Loss curves. Validate the trained model on unseen test images to ensure robustness in challenging environments.

4. Number Plate Extraction: Use YOLO's bounding box output to crop the detected plate region from the vehicle image. Apply preprocessing to the cropped plate such as:

- Grayscale conversion
- Binarization/thresholding
- Noise removal
- Contrast enhancement (CLAHE)
- This step increases OCR accuracy.

5. OCR-Based Character Recognition: Use an OCR engine to read the characters from the extracted plate:

- PaddleOCR (high accuracy, multilingual)
- EasyOCR (fast and simple)
- Tesseract OCR (classical approach, moderate accuracy)
- LPRNet / CRNN / TrOCR (deep-learning-based OCR)

Perform optional steps: Character segmentation (if required), Sequence correction using regex (format-based filtering)

6. Post-Processing: Validate recognized characters using region-specific rules. Remove noisy predictions using confidence-threshold filtering. Combine detection + OCR output to generate the final license plate number.

7. System Integration: Integrate the ANPR pipeline into a real-time system:

- CCTV/Camera input
- Frame-by-frame YOLO detection
- OCR processing
- Database/storage (SQL/MySQL/CSV)

Implement API endpoints for automatic logging, alert generation, and result visualization.

8.Evaluation Metrics: Evaluate system performance using:

- Detection Accuracy (mAP)
- OCR Accuracy (Character-level & Word-level)
- F1-score
- Inference Time (FPS)
- Hardware performance (CPU/GPU/Edge device speed)

9. Deployment: Deploy on selected hardware:

- Laptop/PC
- NVIDIA Jetson Nano / Xavier
- Raspberry Pi
- Cloud (AWS, GCP)

Optimize using TensorRT, ONNX conversion, quantization (INT8/FP16), or model pruning for faster inference.

CONCLUSIONS

The development of an Automatic Number Plate Recognition (ANPR) system using YOLO demonstrates the significant advantages of deep-learning–based approaches over traditional image-processing methods. YOLO's real-time detection capability, combined with robust OCR frameworks, allows the system to accurately identify and extract license plate information even under challenging conditions such as varying illumination, motion blur, shadows, and angle distortions. The integration of preprocessing techniques and optimized OCR pipelines further enhances recognition accuracy across different plate styles and formats. However, the study also highlights persisting challenges, including limited generalization across countries, dataset imbalance, and reduced performance in extreme weather or highly degraded images. Additionally, computational constraints on low-power edge devices remain a key consideration for real-world deployment. Despite these limitations, the proposed approach demonstrates strong potential for use in traffic monitoring, smart parking, toll automation, and security applications. Overall, YOLO-based ANPR provides a fast, reliable, and scalable solution for modern intelligent transportation systems. Future work can focus on extending multilingual support, incorporating explainable AI (XAI), expanding training datasets, and integrating lightweight architectures for edge environments, thereby improving transparency, adaptability, and operational efficiency in practical deployments

REFERENCES

[1] T. M. Al-Hasan, V. Bonnefille and F. Bensaali, "Enhanced YOLOv8-Based System for Automatic Number Plate Recognition," Technologies, vol. 12, no. 9, pp. 1–20, 2024.

[2] B. Buleu, R. Robu and I. Filip, "A Deep Learning-Based System for Automatic License Plate Recognition Using YOLOv12 and PaddleOCR," Applied Sciences, vol. 14, no. 1, pp. 1–18, 2024.

[3] P. Meesad and W. Thumthong, "Advanced Deep Learning Techniques for Automated License Plate Recognition," Scientific Reports, vol. 13, no. 1, pp. 1–18, 2023.

[4] A. Sarhan, R. Abdel-Rahem, B. Darwish, A. Abou-Attia, A. Sneed, S. Hatem, A. Badran and M. Ramadan, "Egyptian Car Plate Recognition Based on YOLOv8, Easy-OCR, and CNN," Journal of Electrical Systems and Information Technology, vol. 11, no. 32, pp. 1–14, 2024.

[5] B. Satya, D. Manongga, Hendry and A. Aminuddin, "Optimized YOLOv8 for Automatic License Plate Recognition on Resource-Constrained Devices," Engineering, Technology & Applied Science Research, vol. 15, no. 2, pp. 21976–21981, 2025.

[6] X. Zhang and S. Yu, "A Lightweight License Plate Recognition Method Based on YOLOv8," Electronics, vol. 14, no. 17, pp. 1–16, 2025.

[7] F. Sonnara, H. Chihaoui and F. Filali, "Efficient Real-Time License Plate Recognition Using Deep Learning on Edge Devices," Journal of Real-Time Image Processing, vol. 22, p. 159, 2025.

[8] D. Sasinath, V. Surya, U. Vaishali and C. Sangeetha, "Survey on Efficient Automated Number Plate Recognition System Using YOLOv8 and Paddle OCR," International Research Journal of Modernization in Engineering Technology and Science, vol. 7, no. 3, Mar. 2025.

[9] P. Pooja, D. Katipamu, L. Kariggi, G. Kuramana, J. Mallireddi and K. Kiranmai, "License Plate Detection and Recognition System Using YOLOv8 and LPRNet," International Journal of

Creative Research Thoughts (IJCRT), vol. 13, no. 4, pp. 1–8, 2025.

[10] G. Kothai, E. P. Ovammal, S. Amutha and V. Deepa, "An Efficient Deep Learning Approach for Automatic License Plate Detection with Novel Feature Extraction," Procedia Computer Science, vol. 235, pp. 2822–2832, 2024.

[11] A. Samaga, A. J. Lobo, A. Nasreen, R. K. Pattar, N. Trivedi, P. Raj and K. Sreelakshmi, "Enhancing Automatic License Plate Recognition in Indian Scenarios," International Journal of Electrical and Computer Engineering (IJECE), vol. 15, no. 1, pp. 365–373, Feb. 2025.

[12] A. Verma, R. Rajbhār, R. Ravi Sahu and R. Sahu, "Real-Time Automatic License Plate Detection Using YOLO for Intelligent Transportation Systems," 2025.

[13] "Real-Time Vehicle License Plate Detection and Recognition Using YOLOv5," Indian Journal of Engineering Research Networking and Development, vol. 2, no. 3, Mar. 2025, Kumar Dadi Institute of Technology.

[14] C. B. Asaju, P. A. Owolawi, C. Tu and E. Van Wyk, "Cloud-Based License Plate Recognition: A Comparative Approach Using You Only Look Once Versions 5, 7, 8, and 9 Object Detection," Information, vol. 16, no. 1, article 57, 2025.

[15] A. Nafis, S. Karn, A. Ali, A. Bora and V. Trivedi, "Automatic Number Plate Recognition Using YOLOv11," International Journal of Computer Techniques, vol. 12, no. 3, pp. 1–6, May–June 2025.