

Article

AVNPR-Net: A Real-Time Deep Learning Framework for Robust Vehicle Number Plate Detection and Recognition

Ahmad Ijaz^{1,*}, Tavyba Sarfraz¹, Tanzeela Bibi², Muhammad Usman³

¹ Department of Computer Science and Technology, School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: officialahmed000@gmail.com (A. Ijaz), tavybasarfraz2001@gmail.com (T. Sarfraz).

² Department of Information and Communication Engineering, School of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: tanzeelabibi26@gmail.com (T. Bibi).

³ Department of Computer Application Technology, School of Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China; e-mail: usman7610@outlook.com (M. Usman).

* Correspondence Author

The authors received no financial support for the research, authorship, and/or publication of this article.

Abstract: AVNPR systems are critical in intelligent transportation, monitoring, and law enforcement systems. Nevertheless, the current systems are usually challenged by the issues of dissimilar illumination, obstruction, and the diversity of plate formats, which restrict their practical applicability. To solve these problems, this paper suggests a real-time deep learning-driven AVNPR framework that incorporates effective detection and recognition systems. The proposed system employs the YOLOv8 object detector model to localize number plates with high accuracy and speed, as well as a lightweight recognition module to identify alphanumeric characters. A custom dataset with different types of vehicles in different environmental conditions was created and improved with the help of preprocessing and data augmentation methods to make the model more robust. In the experiments, the proposed system demonstrated an overall system accuracy of 98.7%, representing the combined number plate detection and character recognition results. The mAP@0.5 is 97%, and mAP 0.5-0.95 is 91%, as well as high precision, recall, and F1-score, which suggests that it shows potential applicability across varying conditions in the assessed dataset and suggests that it may be suitable for real-world applications. The system is also implemented with a Flask-based web application, and it supports image based and real-time webcam detection. The results indicate that the proposed framework provides a viable, efficient, and deployable solution to AVNPR applications. The work will lead to the creation of scalable and real-time intelligent transportation systems and give a basis for future advancement in the improvement of robust vehicle recognition in challenging conditions.

Keywords: Automatic Vehicle Number Plate Recognition; AVNPR; Traffic Surveillance; Law Enforcement; Intelligent Transportation System.

Copyright: © 2026 by the authors. This is an open-access article under the CC-BY-SA license.



1. Introduction

Automatic Vehicle Number Plate Recognition (AVNPR) has become an important part of the current intelligent transportation system and serves to identify vehicles automatically to be used in traffic monitoring, toll collection, parking control, and law enforcement [1]. With the rapid growth of urbanization and the increasing number of vehicles on roads, there is a growing demand for efficient, accurate, and real-time systems capable of identifying vehicle license plates under diverse environmental

conditions [2], [3]. Nevertheless, the practical implementation of AVNPR systems is still difficult because of different illumination, motion blur, occlusion, complicated backgrounds, and non-standard plate formats [4]. The conventional number plate recognition methods were very dependent on rule-based image processing methods, such as edge detection, thresholding, and morphological operations [5], [6]. Although these approaches offered some early solutions, they were inadequate due to their lack of strength and flexibility in changing circumstances [7]. The

shift to machine learning was accompanied by better classification methods, yet these methods still relied on hand-crafted features and multi-stage pipelines [8], [9].

Deep learning-based approaches, especially Convolutional Neural Networks (CNNs), have recently made a tremendous impact on the field, allowing automatic feature extraction and enhanced generalization [10]. The You Only Look Once (YOLO) family of models is one of the most popular deep learning methods because it is capable of detecting objects in real-time with high accuracy [11]. Particularly, YOLOv8 has architectural enhancement, better feature representation, and an effective training process that makes it very suitable in applications where fast and reliable detection is needed [12], [13]. Its single-stage detection architecture enables localization and classification of objects in a single forward scan, thus minimizing computational complexity and latency relative to conventional two-stage detectors [14]. Although these developments exist, most of the current AVNPR systems are mainly concerned with high detection accuracy and ignore practical implementation features like system integration, real-time performance, and usability [15]. Moreover, some of these state-of-the-art methods are based on complicated hybrid architectures and big-scale data that can constrain their use in resource-constrained settings. A trade-off between reliable detection, acceptable recognition performance, and efficient deployment in real-world situations is still needed [16], [17].

In this regard, this paper offers a deep learning-based AVNPR system capable of both vehicle number plate detection and recognition, with the main focus on real-time vehicle number plate detection based on the YOLOv8 model. In addition, the system incorporates a small recognition module to isolate alpha-numeric data of the plates that have been detected and is subsequently implemented in a Flask-based web application to facilitate easy interaction between the user and the system. An experimental dataset with a variety of vehicle types and different environmental conditions is used to increase the generalization of models.

The key contributions of this work are the following:

- Creation of a YOLOv8-based real-time number plate detector that accurately localizes under various environmental conditions.
- Detection and recognition should be integrated into a single lightweight ANPR pipeline, allowing full plate detection.
- Adoption of a Flask web application that facilitates real-time webcam and image-based webcam detection.
- Development of a heterogeneous dataset to enhance the model generalization in different lighting, weather, and vehicle conditions.

The rest of the paper is organized in the following manner. Section 2 conducts a literature review on the field

of automatic vehicle number plate recognition. The proposed methodology, such as dataset preparation, model configuration, and system architecture, is presented in Section 3. The experimental results and performance evaluation are discussed in Section 4. In Section 5, the findings are discussed in detail. Lastly, Sections 6 and 7 wraps up the paper and points out future research directions.

2. Related Work

Automatic Vehicle Number Plate Recognition (AVNPR) is an important application of computer vision that enables the automatic detection and recognition of vehicle license plates [18], [19]. It plays a crucial role in intelligent transportation systems, traffic monitoring, parking management, toll collection, and law enforcement [20]. An AVNPR system is generally made up of several steps, such as image acquisition, preprocessing, plate localization, character segmentation, and optical character recognition (OCR). The early ANPR systems were based on the conventional methods of processing images, including edge detection, thresholding, and morphological operations [21], [22]. Nevertheless, they were very sensitive to environmental factors such as lighting changes, shadows, occlusion, and plate format differences, thereby limiting their practical use in the real world [23]. As the field of artificial intelligence evolves, more modern ANPR systems have been developed using deep learning-based methods to enhance robustness and accuracy [24], [25]. Convolutional Neural Networks (CNNs) have become the backbone of modern AVNPR systems due to their strong feature extraction capabilities [26]. Specifically, plate detection models like Faster R-CNN, SSD, and the YOLO (You Only Look Once) family have become widely popular in plate detection [27]. As single-stage detectors, the YOLO series provides real-time operation by performing object detection and classification in a single step. Recent models such as YOLOv5, YOLOv7, and YOLOv8 included better feature pyramid networks, anchor-free detection, and training efficiency. Moreover, more recent versions like YOLOv9, YOLOv10, YOLOv11, and YOLOv12 have added more sophisticated features like programmable gradients, end-to-end optimization, and hybrid architectures to obtain improved accuracy and scale [28].

Several studies have explored the application of deep learning models in AVNPR systems. N. Ruseno et al. [29] suggested an improved Automatic Vehicle Number Plate Recognition (AVNPR) system to overcome environmental variations, occlusion, and various plate formats in real-time traffic surveillance. The study combines YOLOv8 to identify license plates and a Convolutional Neural Network (CNN) to identify characters, where a large dataset of more than 33,000 images is used to train the model. The system was also very reliable and effective, with 94.4% detection accuracy and 94.05% character recognition accuracy. Although these results are promising, the study also

has limitations such as high computational demands, reliance on large data, and the inability to deal with extremely complex or degraded conditions, which can impact scalability and deployment effectiveness.

S. Arukonda et al. [30] suggested a smart parking system based on AVNPR in real time to solve the problem of inefficient parking distribution and monitoring vehicles in the urban area. The research employs the YOLOv8 to identify license plates and OCR to recognize text, and a rule-based system to assign parking slots. The system had a 93.5% accuracy, 94.2% precision, 95.8% recall, and 94.0% F1-score with a rapid inference of approximately 50 ms/frame, indicating a good real-time performance. However, the research is limited to poor performance in extreme weather conditions, motion blur, non-standard plates, and reliance on image quality, which can influence robustness in a highly dynamic environment.

W. Xiong et al. [31] suggested an improved license plate recognition system that can work under complex conditions with CSCM-YOLOv8 to detect and CSM-LPRNet to recognize. The method enhances YOLOv8 with modules like CPA-Enhancer, CARAFE, SEAM, and MLLABlock to improve feature extraction, robustness, and detection accuracy, while LPRNet is improved using SE, CBAM, and MixConv2d for better character recognition. The system had a 94.50% accuracy, 95.70% precision, 95.8% recall, and 96.0% F1-score, and mAP at 0.50 of 95.9% and 94.9%, respectively, and recognition accuracy of 95.56%, which is much higher than the baseline models. In spite of good performance, the weaknesses are that the models are very complex, expensive to compute, and difficult to implement in real-time on resource-constrained platforms.

A hybrid license plate detection and recognition system for Bangladeshi vehicles was suggested by S. Rahman et al. [32], which overcomes certain issues, such as complex Bangla characters, different lighting, and different plate formats. It is based on YOLOv8 detection and a hybrid recognition method of EfficientNetB0 and EasyOCR, which is combined via a confidence-based voting system. The robustness is enhanced by a multi-stage preprocessing pipeline. The model attained a detection accuracy of 96.7%, 95.3% precision, 95.8% recall, and 94.4% F1-score, and full-plate recognition accuracy of 94.90%, which is better than single CNN and OCR approaches. However, the limitations are computational overhead, reliance on the quality of preprocessing, and worse performance with occlusion, degraded images, and non-standard plates.

V. Gnanaprakash et al. [33] suggested an AVNPR system based on deep learning to track vehicles in real-time with CCTV footage, which solves the issue of tracking vehicles in high-speed conditions. The system scans vehicles and license plates with YOLO, and then processes images with Image processing methods and Tesseract OCR to identify characters, established on the ImageAI frame-

work. The model reached 95.5% car detection, 94% plate detection, and 96.7% character recognition, 96.2% precision, 95.8% recall, and 95.9% F1-score, which shows the successful performance in real-time. The study is limited by poor performance in different lighting conditions, large computational needs, and reliance on the quality of preprocessing, which can influence robustness in the real world.

Other architectures and improvements have also been explored in addition to the YOLO-based approaches. As an example, CNNs have been modified with Recurrent Neural Networks (RNNs) or Connectionist Temporal Classification (CTC) to enhance the accuracy of character recognition [34]. These studies show that although there is a significant improvement in detection accuracy, the overall performance of the system remains a function of the dataset diversity, environmental factors, and the efficiency of the recognition stage. In spite of these developments, there are a number of limitations to the current research. The newer versions of YOLO (v9–v12) have more complex architecture and increased computational demands, and are less applicable to real-time usage in lightweight or web-based applications [35]. Moreover, other designs are mainly concerned with the accuracy of the detection but not the system integration, usability, and efficiency of the deployment. Issues like plate format variability, lighting, and generalization to invisible data are also not yet resolved.

Hence, the current work uses YOLOv8 as the central detection model because it offers a balanced trade-off between accuracy, speed, and the simplicity of implementation. YOLOv8 provides a stable and well-supported framework, enabling efficient training and seamless integration with web technologies such as Flask. It has a reduced computational load compared to more recent versions but still has good detection characteristics, making it very appropriate for real-time AVNPR systems. The goal of the proposed study is to create a practical, efficient, and user-friendly number plate recognition system that can work effectively in the real-world environment.

3. Methods

This study proposes a real-time Automatic Vehicle Number Plate Recognition (AVNPR) system based on the YOLOv8 object detection framework. The system is designed to accurately detect and recognize vehicle number plates under diverse environmental conditions. The overall pipeline consists of dataset preparation, preprocessing, model configuration, training, and deployment within a web-based application.

3.1. Overview of the Proposed System

The proposed study introduces an Automatic Vehicle Number Plate Recognition (AVNPR) system built on deep learning that is capable of real-time and image-based ve-

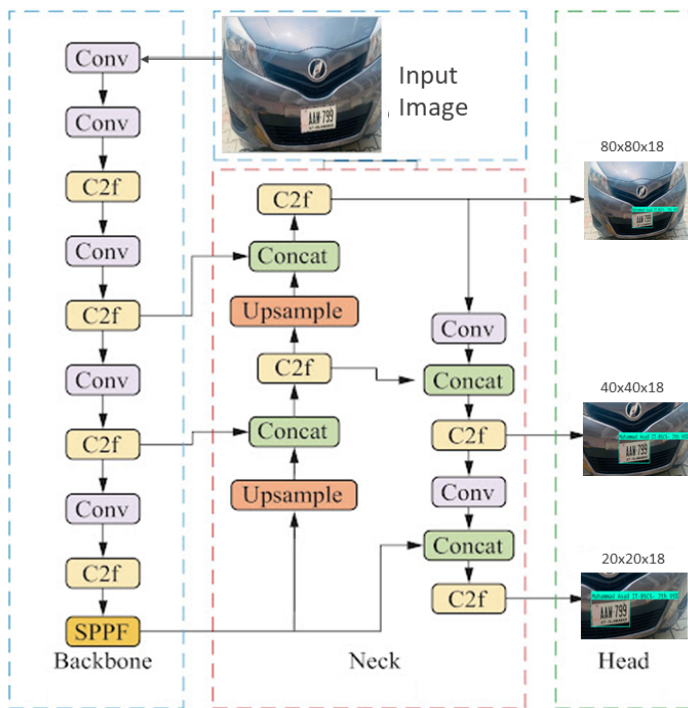


Figure 1. Overview of the Proposed Architecture.

hicle number plate detection and recognition. The system suggested has a systematic pipeline that incorporates data acquisition, preprocessing, model training, detection, recognition, and deployment. YOLOv8 is chosen as compared to previous versions due to its anchor-free architecture, enhanced training process, and generalizability. The system is based on the YOLOv8 object detector model that allows the localization of number plates effectively and in real-time. The plate regions identified undergo further processing to extract and identify characters. In order to boost usability, the trained model is deployed to a Flask-based web app, enabling users to conduct image-based and real-time webcam detection. The system is designed to guarantee robustness, scalability, and applicability in the real world, like traffic monitoring and surveillance systems.

The proposed system has an architectural framework of the YOLOv8 detection pipeline, comprising three primary components: the backbone, neck, and detection head. As shown in the Figure 1, the input image is initially fed into the backbone network, which consists of several convolutional (Conv) and C2f layers to extract hierarchical feature representations, and the Spatial Pyramid Pooling Fast (SPPF) module further improves multi-scale feature learning. These characteristics are subsequently forwarded to the neck that incorporates operations of upsampling and concatenation (Concat) to combine high-level semantic features with low-level spatial features, thus making successful feature merging at varying scales.

Lastly, the detection head uses multi-scale predictions at the various resolutions (e.g., 80×80, 40×40, and 20×20) and enables objects of various sizes to be detected accurately, including small objects like number plates. It is

an effective architecture to use in localizing number plates in complex situations with high precision because it efficiently identifies the plates in real time through the use of robust feature extraction coupled with multi-scale learning.

The suggested methodology makes a clear distinction between the detection and recognition phases. The YOLOv8 model is directly trained to localize number plates, whereas recognition is performed on extracted plate regions. This modular architecture enhances interpretability and scalability, and enables detection and recognition subunits to be optimized separately.

3.2. Data Collection and Preparation

The dataset in this study is a custom dataset constructed to be utilized in the training and testing of the proposed AVNPR system. It comprises a total of 572 annotated images of vehicle number plates, which are gathered in various real-world situations. The data contains the changes in light, weather, viewing angles, and types of vehicles, so that the data is robust and can be generalized. The dataset was broken down into three subsets, 401 images were used to train, 114 images were used to validate, and 57 images were used to test. This division guarantees efficient model learning and steady assessment of unknown data.

The dataset is relatively small compared to large-scale benchmarks; however, its advantage is its variety and variability in the real world. To overcome the weakness of the dataset size, large-scale preprocessing and data augmentation methods were used to artificially enhance the variability of the data and enhance the quality of the models. In addition, a pre-trained YOLOv8 model was used to implement transfer learning, which enables the system to utilize learned representations of larger datasets. This greatly improves detection performance even with small training samples. Consequently, the integration of a curated custom dataset, data augmentation, and transfer learning allows the proposed model to attain strong and trustworthy results in practical settings. Future research will involve the expansion of the dataset to cover larger-scale and multi-region variations in license plates to further improve model generalization. Data augmentation involved random horizontal flips, image resizing, changes to brightness and contrast, and slight rotations. These augmentations were applied during training for real-world variations and increased model generalization.

3.3. Model Training and Configuration

The YOLOv8 model was configured to perform number plate detection as the primary task. Key hyperparameters such as input image size, batch size, learning rate, and number of epochs were carefully selected and optimized based on the dataset characteristics, as shown in Table 1. Transfer learning was applied by ensuring that the

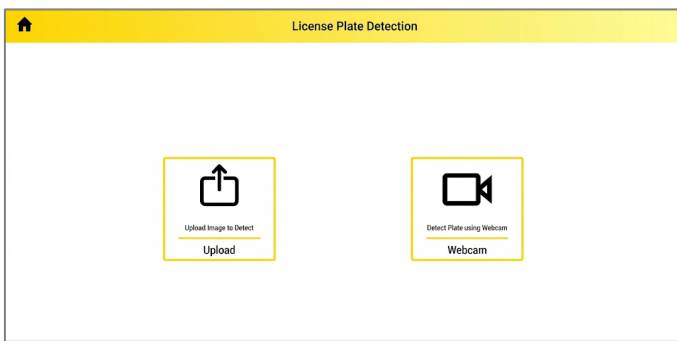


Figure 2. Web Application Interface.

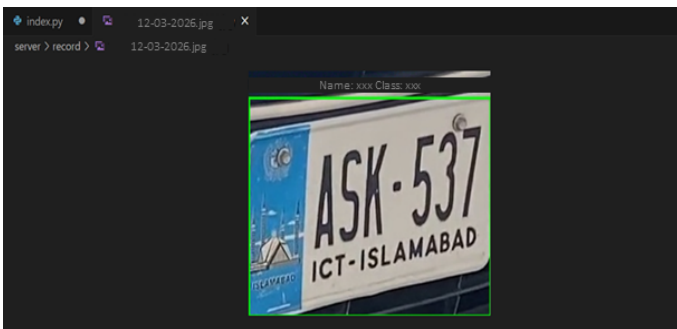


Figure 3. Record Keeping System.

Table 1. Model Configuration and Training Parameters.

| Parameter | Description |
|--------------------|------------------------------------|
| Model Architecture | YOLOv8n (Nano version) |
| Input Image Size | 640 × 640 pixels |
| Number of Epochs | 100 |
| Batch Size | 16 |
| Optimizer | Default (SGD/Adam auto-configured) |
| Hardware Used | Tesla T4 GPU (~15.8 GB VRAM) |
| Model Parameters | ~3.15 Million |
| GFLOPs | ~8.9 GFLOPs |

model was initialized with pre-trained weights that enhanced the speed and effectiveness of training. The model was trained across several epochs, and as the model became capable of learning spatial and contextual features related to number plates. The dataset was divided into training and validation sets to track the performance and avoid overfitting. The optimization was performed based on loss functions of bounding box regression, objectness score, and classification accuracy. Accuracy, precision, and recall were used as performance metrics to measure model effectiveness throughout training. Early stopping and validation checks were implemented to ensure optimal performance while avoiding overfitting.

The final trained model demonstrated the ability to accurately detect number plates under varying environmental conditions and vehicle types. The experiments were carried out on a Tesla T4 GPU with about 15.8 GB of VRAM. Unseen data were used to assess model performance and guarantee generalization. The metrics employed in the evaluation are accuracy, precision, recall,

and F1-score, which give a holistic description of the detection and recognition results. We trained the model with a learning rate of 0.001 and the default optimizer settings of the YOLOv8 framework. Training included data augmentation with horizontal flipping, scaling, brightness, and rotation to improve model robustness. A fixed random seed was used for all experiments to maintain consistency. We used the same data split for all experiments, including ablation studies, to maintain consistency. Training was stopped early according to validation loss to avoid overfitting, and the model with the best validation accuracy was chosen.

3.4. Number Plate Detection and Recognition

3.4.1. Plate Detection

The plate detection phase is carried out with the help of the trained YOLOv8 model, which is based on a single-stage object recognition mechanism that recognizes number plate areas in images or video frames. The model will process the input image and will produce bounding boxes around the identified plates and confidence scores. YOLOv8 can effectively detect small objects and is capable of detecting number plates in complex background clutter, such as poor lighting, occlusion, and arbitrary rotation. The architecture uses an anchor-free detection approach with a feature pyramid to effectively capture multi-scale features and enhance object localization accuracy regardless of object size. This design ensures fast and robust detection in various conditions.

3.4.2. Plate Recognition

When the region of the number plate is identified, the character recognition is performed by extracting the number plate and processing it. The recognition phase is based on the use of image processing to refine the plate area. The stage is especially difficult because of differences in plate formats, fonts, and distortions caused by the environment. The system uses the learned characteristics during the training stage to correctly identify characters, so that even non-ideal conditions will yield reliable results.

While the recognition module is simpler than intricate deep learning-based OCR systems, the quality of the detection stage enhances its performance. Because the YOLOv8 model provides precise localization of number plate zones with minimal distractions, the recognition module receives clean and well-cropped images, enhancing its effectiveness. This approach allows us to achieve effective recognition without a sophisticated recognition module.

3.5. System deployment with Flask Web Application

The trained AVNPR model was implemented into a Flask-based web application to enable its usability in the real world. Flask was selected because it is a lightweight architecture and can easily integrate ML models.

As illustrated in Figure 2, the web application offers two main functions: Image-based detection, where users upload images containing vehicles, and real-time detection, in which the system processes live video input from a webcam. The image processing, model inference, and generation of results are done in the backend, which is based on the Flask implementation. The frontend is created with HTML, CSS, and JavaScript, which include an interactive interface that allows users to interact with the front end. Other libraries like OpenCV, NumPy, and PIL are employed in image handling and processing.

A record management feature is also included in the system, where identified number plates are kept with time stamps to be referred to in the future, as shown in Figure 3. This integration guarantees that it is an end-to-end solution, which comprises detection, recognition, and deployment in a single and unified system that can be used practically in areas like traffic monitoring, parking control, and law enforcement.

3.6. Evaluation Metrics

In order to assess the performance of the proposed AVNPR system quantitatively, various standard evaluation measures, such as accuracy, precision, recall, and F1-score, were applied. Besides classification-based evaluation metrics, object detection models are evaluated using mean Average Precision (mAP), which is the standard metric used for YOLO models. The mAP@0.5 is the average precision with an Intersection over Union (IoU) of 0.5, while the mAP@0.5:0.95 is the average precision with IoU levels ranging between 0.5 and 0.95. This offers a holistic assessment of object detection and localization. These measures are obtained based on the confusion matrix, comprising true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Accuracy indicates how correct a model is in general, and is defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision measures the percentage of number plates identified correctly out of all the number plates identified:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall (Sensitivity) evaluates how well the model can identify all relevant number plates:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score gives a tradeoff between precision and recall:

$$F1 = 2x \frac{(Precision \times Recall)}{Precision + Recall} \quad (4)$$

These metrics offer an overall assessment of the detection and recognition capabilities of the proposed system. It's worth noting that the proposed system is an object detection system rather than a classification system. The confusion matrix and other performance metrics are calculated based on detection performance with respect to predicted bounding boxes and their associated ground truth annotations. A bounding box is classified as correct if the Intersection over Union (IoU) with the ground truth bounding box is greater than a certain threshold (e.g., IoU ≥ 0.5). Using this threshold, true positives (TP), false positives (FP), and false negatives (FN) are determined to compute the metrics. This method allows the computation of object detection evaluation metrics.

4. Experimental Results and Analysis

4.1. Overall Performance Evaluation

The AVNPR system was tested on a private dataset of 401 training, 114 validation, and 57 test images. The overall accuracy of the system was measured at 98.7%, combining the results of number plate detection using YOLOv8 and recognition. This accuracy is largely attributable to the effectiveness of YOLOv8-based detection, which accurately locates the number plate and facilitates recognition of individual characters. The detection results were also confirmed using the standard object detection metrics, with 97% mAP@0.5 and 91% mAP@0.5:0.95, showing strong localization capabilities at different IoU levels. The recognition module was also tested independently (Section 4.7) to verify its effectiveness while being compact. The model exhibited fast inference speeds with minimal latency, although there was some performance decline under difficult conditions such as poor lighting and background clutter.

Figure 4 shows the confusion matrix of the detection results, where the detected number plates are compared to the ground truth with an IoU threshold (IoU ≥ 0.5). True positives are the number of correctly detected number plates, whereas false positives and false negatives are errors and omissions, respectively. The high values along the diagonal of the matrix demonstrate accurate and reliable detection performance. These results demonstrate that the proposed system works well under different conditions in the dataset; however, as evaluation was done on a single dataset, additional testing on other datasets is needed to further validate its effectiveness in real-world conditions.

4.2. Accuracy

The stated accuracy is calculated using detection output results, with correct number plate localization (calculated using IoU) and correct number plate recognition

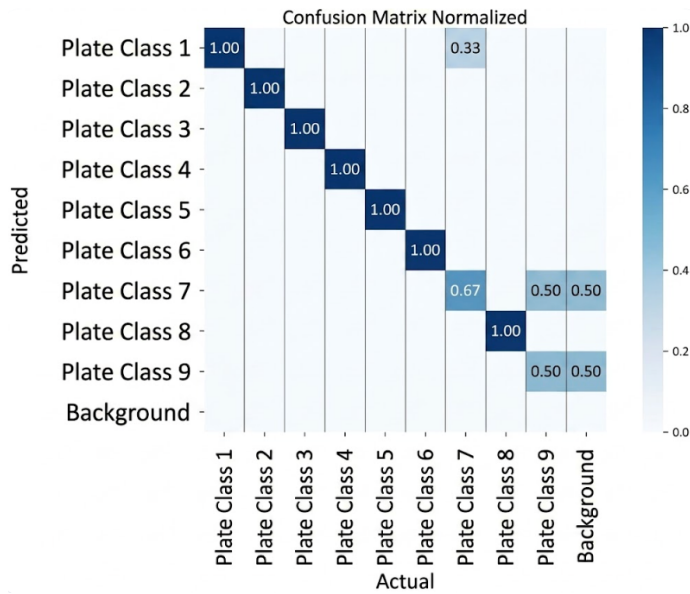


Figure 4. Normalized Confusion Matrix for License Plate Detection.

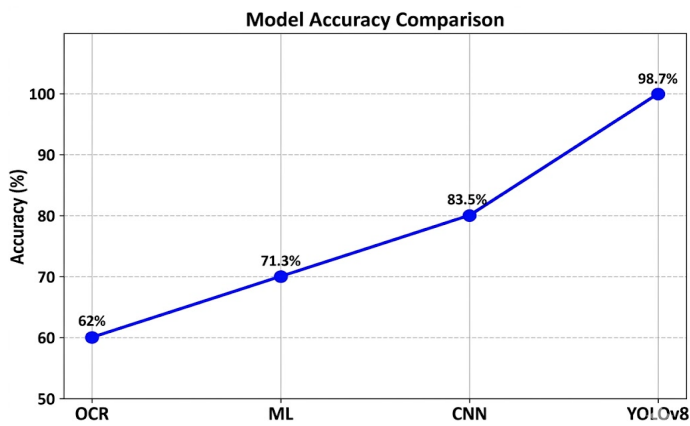


Figure 5. Model Accuracy Comparison.

taken as true positives. The main measure of assessing the general correctness of the system is accuracy, which determines the percentage of correctly detected and recognized number plates in relation to the total number of samples. As shown in Figure 5, the accuracy of the proposed system is compared with existing techniques indicates that the system was effective in dealing with real-world variations like lighting variations, plate orientation, and background noise. The accuracy value reported here corresponds to the overall system performance, including both detection and recognition stages of the AVNPR system, and demonstrates that the YOLOv8 model is capable of learning robust features for number plate detection, although further improvements can be achieved with larger and more diverse datasets.

4.3. Precision

The precision is computed based on detection results, where number plates localized correctly are considered true positives, and incorrect or false detections are treated as false positives. Precision is a measure of how many

number plates are correctly identified, as compared to the total number of number plates detected, and can be used to show how the system detects only false positives. The proposed system had a high value of precision, and indicated that a majority of the number plates detected were classified correctly, as shown in Figure 6. This implies that the YOLOv8 model is effective in minimizing false detections, so that irrelevant areas are not recognized as number plates. The use of high precision is especially vital in the case of law enforcement and surveillance, where a wrong identification can result in serious mistakes.

4.4. Recall

Recall reflects the capability of the detection model to detect all number plate instances, based on bounding box matching with the ground truth labels. The results of the experiment show that the system has a satisfactory recall value and thus it is able to identify most number plates under various circumstances, as shown in Figure 7. However, some missed detections were observed in challenging scenarios, such as low-resolution images, motion blur, and extreme lighting conditions. These limitations highlight the need for further optimization in handling edge cases.

4.5. F1-Score

The F1-score is the harmonic mean of both precision and recall, capturing the overall performance of the detection model. As shown in Figure 8, the suggested AVNPR system had a competitive F1-score, suggesting that there was a good trade-off between the accuracy and completeness of detection. The equal performance indicates that the system is consistent in reducing false positives as well as maximizing true detections. This measure is a validation of the general stability of the system when it is used in the real world, where precision and recall are essential.

4.6. mAP Evaluation (mAP@0.5 and mAP@0.5:0.95)

To assess the performance of the proposed AVNPR system, we also adopted conventional object detection evaluation metrics, such as mean Average Precision (mAP). Specifically, mAP@0.5 and mAP@0.5:0.95 were employed to evaluate the performance of the proposed YOLOv8-based detection model on the basis of different Intersection over Union (IoU) criteria. The proposed system attained an mAP@0.5 score of 97% and an mAP@0.5:0.95 of 91%, demonstrating high detection and localization accuracy. The strong mAP@0.5 score indicates the model's ability to accurately localize number plate regions when using a moderate level of overlap, while the mAP@0.5:0.95 value suggests reliable performance when a higher degree of overlap is required. As illustrated in Figure 9, both metrics show stable convergence during training, which suggests the stability and efficacy of the proposed model in practical applications.

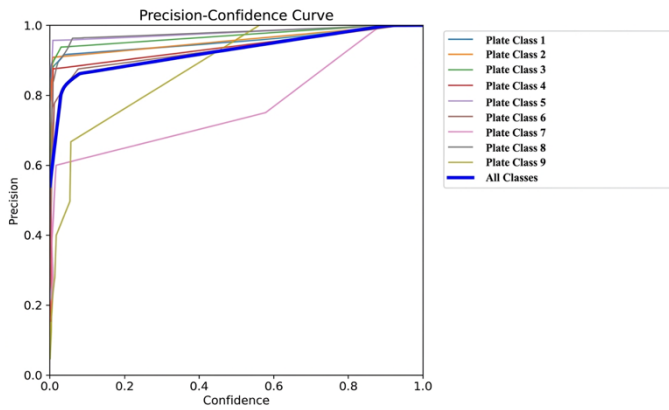


Figure 6. Precision Curve.

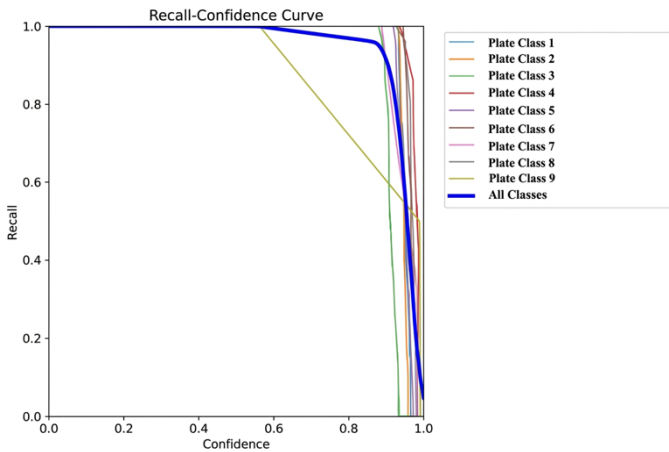


Figure 7. Recall Curve.

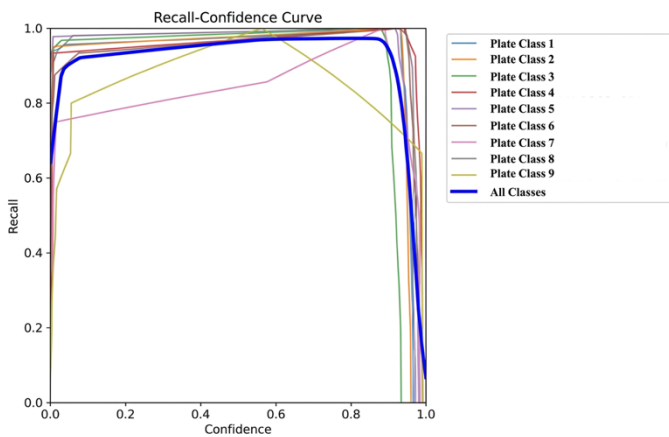


Figure 8. F1-Score Curve.

Table 2. Recognition Performance Metrics of the Proposed AVNPR System.

| Metric | Value (%) |
|--------------------------|-----------|
| Character Level Accuracy | 96.5 |
| Plate Level Accuracy | 92.3 |
| Recognition Precision | 95.8 |
| Recognition Recall | 95.1 |
| Recognition F1-Score | 95.4 |

4.7. Recognition Performance Evaluation

To demonstrate the effectiveness of the proposed AVNPR system, the recognition module was evaluated independently. The performance was evaluated by the correct extraction of the number plate's alphanumeric characters. Two types of accuracy were evaluated: character-level and plate-level accuracy. Character-level accuracy refers to the percentage of characters correctly extracted from all number plates, and plate-level accuracy indicates whether the whole number plate string has been correctly extracted. The system achieves a character-level accuracy of around 96.5% and a plate-level accuracy of 92.3%, as demonstrated in Table 2.

These measures show that while the recognition module is straightforward, it is effective because of the quality of the input it receives from the detection stage, which accurately locates the number plate and extracts it with minimal background noise.

4.8. Real-Time Performance Evaluation

As shown in Table 3, the real-time performance evaluation shows that the suggested AVNPR system satisfies real-time application requirements by achieving effective inference. The real-time inference assessment demonstrates that the AVNPR system performs inference efficiently with an average detection time of 45-60 ms per frame, and an overall end-to-end processing time (including both detection and recognition) of 50-65 ms per frame. The separation of detection time and end-to-end processing time helps to understand the latency of the system. The system is appropriate for implementation in web-based applications since the lightweight YOLOv8 model guarantees minimal latency and moderate resource usage. The system's practical usability in real-world applications is ensured by the integration with a Flask application, which further validates its capacity to process both image-based and live video inputs with low delay.

4.9. Ablation Study

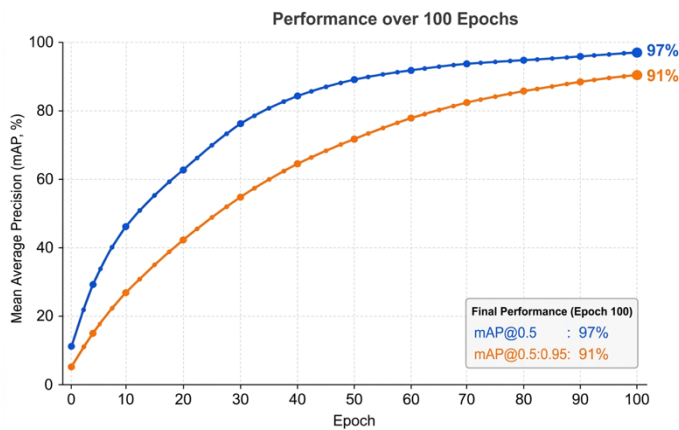
To evaluate the contribution of individual components in the proposed Automatic Vehicle Number Plate Recognition (AVNPR) system, an ablation study was conducted by systematically modifying key elements of the framework. The objective of this analysis was to assess how preprocessing and data augmentation influence the overall performance of the YOLOv8-based model. The accuracy reported in this section refers to the overall system performance, including both detection and recognition. The model trained on raw data without data augmentation and preprocessing achieved an accuracy of around 91.8%, demonstrating poor generalization. The addition of data augmentation resulted in an accuracy of 93.5%, confirming its contribution to generalization performance. Likewise, the use of preprocessing led to an increase in accuracy to 95.7%, demonstrating the benefits of input qual-

Table 3. Real-time Performance Metrics of the Proposed Study.

| Metric | Value / Description |
|----------------------------|----------------------------------|
| Inference Time (per frame) | ~45–60 ms |
| Frames Per Second (FPS) | ~16–22 FPS |
| Detection Time | ~30–40 ms |
| Recognition Time | ~10–15 ms |
| End-to-End Processing Time | ~50–65 ms |
| Model Size | ~6–8 MB (YOLOv8n) |
| GPU Used | Tesla T4 (~15.8 GB VRAM) |
| GPU Memory Usage | ~2.1–2.3 GB |
| CPU Usage | Moderate (depends on deployment) |
| Latency | Low (near real-time response) |
| Deployment Platform | Flask Web Application |
| Input Mode | Image Upload + Real-Time Webcam |

Table 4. Comparative Analysis of Existing AVNPR Studies with Proposed Framework.

| Authors | Techniques/Models Used | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|-----------------------------|---------------------------|--------------|---------------|------------|--------------|
| N. Ruseno et al. [29] | YOLOv8 + EasyOCR | 94.4 | x | x | x |
| S. Arukonda et al. [30] | YOLOv8 + OCR | 93.5 | 94.2 | 95.8 | 94.0 |
| W. Xiong et al. [31] | YOLOv8 + Semi-supervised | 94.5 | 95.7 | 95.8 | 96.0 |
| S. Rahman et al. [32] | YOLOv8 + EasyOCR | 96.7 | 95.3 | 95.8 | 94.4 |
| V. Gnanaprakash et al. [33] | YOLOv8 | 94.0 | 96.2 | 95.8 | 95.9 |
| M.R. Alam et al. [34] | YOLOv8 + Preprocessing | 93.7 | 91.3 | 95.7 | x |
| S. Basak et al. [35] | CNN / ML-based ANPR | 96.9 | x | x | x |
| Proposed Study | YOLOv8 + Flask Deployment | 98.7 | 97.6 | 97.1 | 95.3 |

**Figure 9.** mAP@0.5 and mAP@0.5:0.95 performance curves.

ity improvement. Lastly, the use of both preprocessing and data augmentation led to the highest accuracy of 98.7%, confirming the effectiveness of the integrated approach.

Conversely, the proposed system, based on YOLOv8, has an efficient accuracy of 98.7% and has a balanced performance in precision, recall, and F1-score. The main benefit of the given solution is that it is simple, lightweight, and can be deployed in real-time with a Flask web application, while also achieving higher accuracy compared to existing approaches. As demonstrated in Table 4, the proposed method achieves the highest

accuracy among all compared approaches. Image preprocessing enhances image quality and reduces noise, leading the detection model to learn better features. Augmentation enhances the diversity of the data, allowing the model to adapt to different environmental factors.

To confirm the effectiveness of these enhancements, we trained and tested all models under the same experimental conditions using the same data split. Testing was always done on the same set of unseen images, ensuring that comparisons between configurations were fair. This confirms that the observed performance gain is systematic and not due to experimental variation or bias. Ablation studies were performed using the same training settings, such as hyperparameters, data split, and evaluation metrics. This allows us to compare the performance gains from incorporating preprocessing and data augmentation, rather than differences in the experimental conditions.

5. Discussion

The evaluation shows that the proposed AVNPR system performs well by leveraging the integration of a YOLOv8-based object detector and a lightweight recognizer. The overall high accuracy is largely attributable to accurate detection, allowing the recognition component to work on clear and well-cropped inputs. Custom data, data augmentation, and pre-trained models help achieve better generalization even with a small dataset. The evaluation is

based on object detection principles with IoU matching, and the high mAP@0.5 and mAP@0.5:0.95 indicate the detection model's effectiveness. The independent evaluation of character and plate accuracy also confirms that the system is capable of an end-to-end AVNPR system. However, the robustness claim is limited, and additional evaluations on other datasets are needed to validate the model's generalization. While the system performs well, some failures are noted. False positives may arise when text-like objects are confused with number plates, while false negatives can be caused by harsh conditions such as low brightness, blur, occlusion, and low resolution. Recognition errors may also occur when detected plate regions are partially distorted. Such shortcomings suggest the need for more diverse data, better preprocessing, and more effective recognition methods. While the proposed system delivers higher accuracy and efficiency than other methods, baseline comparisons with the same data under the same model settings were not undertaken. Cross-dataset testing and baseline comparisons will be conducted in future work to further demonstrate the effectiveness and robustness of the proposed system.

6. Limitations and Future Work

Although the proposed AVNPR system has shown promising performance, some limitations still exist. Compared to the other modules, the recognition module is relatively simple and might not be optimal in some demanding environments like low-resolution images, occlusion, and non-standard plate formats. Moreover, the dataset size and diversity can constrain the extrapolation of the model to unseen environments. The future work will be aimed at incorporating new recognition methods, such as OCR-based or sequence learning models, to enhance the

accuracy of character recognition. Moreover, increasing the dataset to more realistic life situations and streamlining the model to be used on the edge will improve the stability and usability of the system. The lack of cross-dataset evaluation is another limitation of the current study. The model has only been tested on a custom dataset, which may not capture all real-world scenarios. In this regard, future research should aim to test the model on publicly available datasets and cross-dataset scenarios for robust generalization capabilities.

7. Conclusion

This paper introduced an Automatic Vehicle Number Plate Recognition (AVNPR) system that uses deep learning and combines the use of YOLOv8 to detect the number plates in real-time with the lightweight recognition model and a Flask-based web app to implement it in practice. The proposed system proves to be highly efficient with an overall accuracy of 98.7% and strong robustness under diverse environmental conditions within the evaluated dataset, including light, weather, and diversity of vehicles. The system effectively addresses the existing gap between model development and real-life implementation by offering image-based and real-time detection features in an easy-to-use interface. Although the recognition component remains relatively simple and performance can be further improved through enhanced datasets and advanced OCR techniques, the proposed framework offers a lightweight, efficient, and deployable solution. In general, the paper helps to develop practical AVNPR systems and serves as a platform upon which future research in the field of intelligent transportation and surveillance applications can be conducted.

8. Declarations

8.1. Author Contributions

Ahmad Ijaz: Conceptualization, Methodology, Writing Original Draft; **Tayyba Sarfraz:** Conceptual Guidance, Critical Review, Formal Analysis; **Tanzeela Bibi:** Data Visualization, Formal Analysis, Investigation and Review; **Muhammad Usman:** Data Collection, Software Validation and Critical Review.

8.2. Institutional Review Board Statement

Not applicable.

8.3. Informed Consent Statement

Not applicable.

8.4. Data Availability Statement

The data used in this study is available on the Roboflow platform via the following link: <https://app.roboflow.com/ahmed-xd-di5fk/anpr-zp1fg/models>.

8.5. Acknowledgment

Not applicable.

8.6. Conflicts of Interest

The authors declare no conflict of interest.

9. References

- [1] M. Shawon, S. Molla, M. S. Nowjh, A. E. Emon, and K. Fatema, "Automatic number plate recognition (ANPR) system enhancing security in Bangladesh," *Journal of Trends and Challenges in Artificial Intelligence*, vol. 3, no. 2, pp. 83-92, 2026. <https://jai.aspur.rs/archive/v3/n2/3.pdf>.
- [2] T. Sampath and R. Kulandaivel, "Advanced Real-Time License Plate Recognition Using Modified Lightweight Convolutional Greylag Goose Network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 40, no. 3, p. 2550038, 2026. <https://doi.org/10.1142/S0218001425500387>.
- [3] O. Bol, B. Kem, and S. Sin, "Automatic Cambodian License Plate Recognition Using Deep Learning With Robust Skew Handling," *AIJR Proceedings*, vol. 8, no. 1, pp. 22-30, 2026. <https://doi.org/10.21467/proceedings.8.1.4>.
- [4] S. Sravanth, E. Shamitha, M. V. Sai, and Y. Raju, "An Automated Challan Generation System Leveraging Deep Neural Networks for Precise Violation Detection and License Plate Recognition," *Digital Cities*, pp. 449-460, 2026. <https://doi.org/10.1002/9781394233823.ch29>.
- [5] W. Gata, M. Haris, M. I. Prasetiyowati, and S. Harianto, "EagleEyes: An Artificial Intelligence-Based Approach for Automatic Traffic Violation Detection Using Deep Learning," *Journal of Applied Data Sciences*, vol. 7, no. 1, pp. 277-289, 2026. <https://doi.org/10.47738/jads.v7i1.1097>.
- [6] M. Z. Hasan and S. G. S. Al-Ali, "Detection and Recognition for Iraqi Modern License Plate Using Deep Learning Approach," *Statistics, Optimization and Information Computing*, vol. 15, no. 1, pp. 771-786, 2026. <https://doi.org/10.19139/soic-2310-5070-2801>.
- [7] K. R. Singh, R. S. Khedgaonkar, P. M. Sonsare, K. Gupta, and S. S. Badhiye, "Automatic Helmet Detection from Videos Using Artificial Intelligence Method and Engage with Secured Driving System by Message Alert," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, pp. 1-17, 2026. <https://doi.org/10.1007/s40998-025-00963-7>.
- [8] J. Kang et al., "KangNet: Compact Deblurring Model with GAN-Based Blur Translation for License Plate Recognition," *Journal of Information Science and Engineering*, vol. 42, pp. 213-226, 2026. [https://doi.org/10.6688/JISE.202601_42\(1\).0012](https://doi.org/10.6688/JISE.202601_42(1).0012).
- [9] X. Fan and W. Zhao, "A Robust Multi-Oriented License Plate Detector and A Derived End-to-End License Plate Recognizer," *IET Image Processing*, vol. 20, no. 1, p. e70272, 2026. <https://doi.org/10.1049/ipr2.70272>.
- [10] X. Huang, H. Zhang, J. Ge, and W. Tang, "YOLO-B2DV: An Integrated Real-Time Framework for Detection, Tracking and Recognition of Catenary Pole Number Plates in Railway Video Streams," *IEEE Access*, vol. 14, 2026. <https://doi.org/10.1109/ACCESS.2026.3653958>.
- [11] Q. Chen et al., "Automatic Real-Time Queue Length Detection Method of Multiple Lanes at Intersections Based on Roadside LiDAR," *Electronics*, vol. 15, no. 3, p. 585, 2026. <https://doi.org/10.3390/electronics15030585>.
- [12] G. B. Nardes, T. H. Rausch, B. H. Pereira, D. R. Melo, and C. A. Zeferino, "A resource-constrained CNN accelerator for real-time license plate character recognition on FPGA platforms," *Integration*, vol. 108, p. 102654, 2026. <https://doi.org/10.1016/j.vlsi.2026.102654>.
- [13] S-L. Lin and C-W. Li, "Research on Integrating Deep Learning-Based Vehicle Brand and Model Recognition into a Police Intelligence Analysis Platform," *Computers, Materials, and Continua*, vol. 86, no. 2, p. 1, 2026. <https://doi.org/10.32604/cmc.2025.071915>.
- [14] M. S. Hanif, Z. Nawaz, and M. K. Malik, "Smart Vision Traffic Surveillance: Vehicle Re-Identification and Tracking Using Vision Transformer," *Vehicles*, vol. 8, no. 2, p. 36, 2026. <https://doi.org/10.3390/vehicles8020036>.
- [15] A. Elouali and A. J. Jara, "Fine-Grained Vehicle Make and Model Recognition for Smart City Environmental Monitoring: A YOLO11-Based Two-Stage Framework," *Urban Science*, vol. 10, no. 2, p. 74, 2026. <https://doi.org/10.3390/urbansci10020074>.
- [16] B. Narsingoju and R. Jain, "Optimizing Slot Detection by Using Advanced Object Detection Techniques for Intelligent Parking Solutions," *Engineering, Technology and Applied Science Research*, vol. 16, no. 1, pp. 32298-32304, 2026. <https://doi.org/10.48084/etasr.11317>.
- [17] Y. Jebraaily, Y. Sharafi, M. Teshnehlav, and N. A. Ramezanloo, "An optimized hybrid framework for car theft detection: comparative insights from deep transfer learning and

- feature-based machine learning," *Artificial Intelligence Review*, vol. 59, no. 2, p. 68, 2026. <https://doi.org/10.1007/s10462-025-11480-8>.
- [18] M. Chaman, A. El Maliki, Y. Natij, H. Dahou, and A. Hadjoudja, "Deep learning-based real time speed limit sign detection with YOLOv12 on edge AI platforms for embedded ADAS," *Bulletin of Electrical Engineering and Informatics*, vol. 15, no. 1, pp. 396-405, 2026. <https://doi.org/10.11591/eei.v15i1.11128>.
- [19] Q. Chen, Y. Zhao, X. Luo, W. He, and G. Shi, "Hybrid Deep Learning and Classification Framework for Automatic Traffic Inspection Classification Based on Image Detection," *Transactions on Emerging Telecommunications Technologies*, vol. 37, no. 2, p. e70325, 2026. <https://doi.org/10.1002/ett.70325>.
- [20] P. Pokorny, S. H. Berge, T. De Jong, C. George, and T. Bjørnskau, "Right-of-way interactions between automated vehicles and other road users in winter conditions—Insights from Oslo, Norway," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 116, p. 103443, 2026. <https://doi.org/10.1016/j.trf.2025.103443>.
- [21] E. Sifuentes, G. Bravo, F. Enríquez-Aguilera, and A. Mejía-Aguilar, "Technologies for Vehicle Classification in Intelligent Transportation Systems: Advances, Challenges, and Perspectives," *Decision Support Systems for Industrial and Scientific Applications*, pp. 3-22, 2026. https://doi.org/10.1007/978-3-031-89700-9_1.
- [22] M. A. Rahmat and A. I. M. Lukman, "Optimizing Lane Detection in Autonomous Vehicles Using Cascading Attention Mechanisms in DeepLabv3+," *Engineering, Technology and Applied Science Research*, vol. 16, no. 1, pp. 31437-31448, 2026. <https://doi.org/10.48084/etasr.15315>.
- [23] R. Mahum, U. Irshad, A. M. El-Sherbeeney, M. Shehab, and S. M. Lee, "YOLO-O: An Improved YOLO-Based Framework for Vehicle Detection," *IET Image Processing*, vol. 20, no. 1, p. e70337, 2026. <https://doi.org/10.1049/ipr2.70337>.
- [24] J. Nadaf, A. Kadam, P. Jadhav, P. Kadam, V. Patil, and M. Gaikwad, "Vehicle registration plate recognition using YOLOv8 and IoT," in *2025 1st International Conference on AIML-Applications for Engineering & Technology (ICAET)*, 2025, pp. 1-6: IEEE. <https://doi.org/10.1109/ICAET63349.2025.10932268>.
- [25] L. Z. You, B. Z. Jun, P. Y. Han, O. S. Yin, K. W. How, and H. F. San, "Integration of Yolov8 and Paddleocr for car plate recognition system in diverse environments: Malaysia," *Journal of Engineering Science and Technology*, vol. 20, no. 1, pp. 250-270, 2025. https://jestec.taylors.edu.my/Vol%202020%20Issue%201%20February%202025/20_1_16.pdf.
- [26] B. Ramasubramanian, D. Gayathri, K. Priyadharshini, S. Mirdula, P. Sudhakaran, and S. Sabapathy, "Automatic Number Plate Recognition using Vision Transformers and YOLOv8 for ITS," in *2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2025, pp. 1288-1292. <https://doi.org/10.1109/ICESC65114.2025.11212618>.
- [27] A. Al Noman, N. J. Mila, A. H. Bornil, A. Mazumder, N. Khan, and M. H. I. Bijoy, "BanglaPlate: Automated Vehicle Number Plate Detection and Recognition using YOLOv8," in *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2025, pp. 1-6: IEEE. <https://doi.org/10.1109/ECCE64574.2025.11013529>.
- [28] M. Meghana and P. Spurthy, "YOLOv8-Paddle OCR Model for Car Model and Automatic Number Plate Recognition," in *2025 IEEE 4th International Conference for Advancement in Technology (ICONAT)*, 2025, pp. 1-7. <https://doi.org/10.1109/ICONAT66879.2025.11362615>.
- [29] N. Ruseno, F. Ernawan, M. A. Salamat, M. S. Arshad, and H. Amnur, "An Enhanced YOLOv8 Model Integrated with CNN for Real-Time Vehicle License Plate Recognition," *International Journal on Informatics Visualization*, vol. 10, no. 1, pp. 343-349, 2026. <https://dx.doi.org/10.62527/joiv.10.1.5145>.
- [30] S. Arukonda, G. S. Jayanth, A. S. S. Koushik, T. Sarupya, P. V. Kumar, and K. B. Reddy, "Real-Time Vehicle Number Plate Recognition and Smart Parking Allocation Using YOLOv8 and OCR for Intelligent Urban Mobility," *International Journal of Intelligent Transportation Systems Research*, pp. 1-13, 2026. <https://doi.org/10.1007/s13177-025-00612-7>.
- [31] W. Xiong et al., "License plate recognition methodology in complex scenarios based on CSM-YOLOv8 and CSM-LPRNet," *PLoS One*, vol. 21, no. 1, p. e0339649, 2026. <https://doi.org/10.1371/journal.pone.0339649>.
- [32] S. Rahman, M. S. Al Ahsan, S. Adhikary, and M. J. Uddin, "License Plate Detection and Recognition for Bangladeshi Vehicles," in *2026 5th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, 2026, pp. 1-6: IEEE. <https://doi.org/10.1109/ICECTE69292.2026.11429408>.

- [33] V. Gnanaprakash, N. Kanthimathi, and N. Saranya, "Automatic number plate recognition using deep learning," in *IOP Conference series: materials science and engineering*, 2021, vol. 1084, no. 1, p. 012027: IOP Publishing. <http://doi.org/10.1088/1757-899X/1084/1/012027>.
- [34] M. R. Alam, N. Biswas, and M. Z. Mahmud, "Advanced Traffic Optimization and Automated Toll Collection System using Automated Number Plate Recognition and Modified YOLO Algorithm," in *2025 IEEE 2nd International Conference on Computing, Applications and Systems (COMPAS)*, 2025, pp. 1-6: IEEE. <https://doi.org/10.1109/COMPAS67506.2025.11381723>.
- [35] S. Basak, A. Khan, J. Tulswani, U. Jangir, and A. Yadav, "Automated Overspeed Vehicle Detection and Number Plate Recognition Using Deep Learning," in *2025 13th International Conference on Intelligent Embedded, MicroElectronics, Communication and Optical Networks (IEMECON)*, 2025, pp. 1-5: IEEE. <https://doi.org/10.1109/IEMECON69302.2025.11365850>.