Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Real-Time Vehicle Number Plate Detection and Recognition Using YOLOv5 and OCR

Naga Visala Sreeja Akella

Under Graduate, Department of CSE, Neil Gogte Institute of Technology, Hyderabad, Telangana

Correspondence

Naga Visala Sreeja Akella

Under graduate, Department of CSE, Neil Gogte Institute of Technology, Hyderabad, Telangana

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Keywords

Vehicle number plate detection, deep learning, CNN, OCR, intelligent transportation systems

Abstract

The detection of vehicle number plates is a critical component in various applications, including traffic management, law enforcement, and automated toll collection systems. Traditional methods for number plate detection often rely on manual intervention or simple image processing techniques, which can be time-consuming and prone to errors. This project aims to leverage the power of deep learning to develop an automated and efficient system for detecting vehicle number plates.

Deep learning, a subset of machine learning, has revolutionized the field of computer vision by enabling the creation of models that can learn and make decisions based on large datasets. Convolutional Neural Networks (CNNs), a type of deep learning model, are particularly well-suited for image recognition tasks. By training a CNN on a diverse dataset of vehicle images, the model can learn to identify and extract number plates with high accuracy.

The proposed system will consist of several key components: image acquisition, preprocessing, number plate detection, and recognition. Image acquisition involves capturing high-quality images of vehicles from various angles and distances. Preprocessing steps, such as resizing, normalization, and noise reduction, will ensure that the input images are suitable for the deep learning model. The number plate detection phase will employ a CNN to locate the number plate within the image. Finally, the recognition phase will use Optical Character Recognition (OCR) techniques to extract and interpret the characters on the number plate.

The primary objectives of this project are to develop a robust and accurate number plate detection system, evaluate its performance using standard metrics, and demonstrate its practical applicability in real-world scenarios. By addressing the challenges associated with varying lighting conditions, occlusions, and different vehicle types, this project seeks to contribute to the advancement of intelligent transportation systems.

Introduction

Automatic Number Plate Recognition (ANPR) systems have become a crucial part of modern intelligent transportation infrastructure. These systems are widely used for applications such as traffic law enforcement, toll collection, parking access control, and vehicle tracking. Traditional ANPR systems primarily rely on classical image processing techniques such as edge detection, morphological operations, and template matching, which are often sensitive to noise, lighting variations, and orientation changes. As a result, their performance is significantly limited in real-world conditions [1].

The evolution of deep learning, particularly Convolutional Neural Networks (CNNs), has transformed the capabilities of computer vision systems. CNNs have proven effective in tasks such as object detection, segmentation, and character recognition, due to their ability to learn robust, hierarchical features from large datasets [2,3]. These advancements have paved the way for more reliable and scalable ANPR systems that can operate in unconstrained environments such as highways, urban traffic, and nighttime surveillance [4,5].

In a typical deep learning-based ANPR pipeline, vehicle images are first captured using static or mobile cameras. The number plate is then localized using object detection models such as YOLOv3/v4 or Faster R-CNN [6,7]. After detection, Optical Character Recognition (OCR) techniques—such as Tesseract or CRNN—are applied to decode the alphanumeric characters on the plate [8,9].

This research proposes an end-to-end deep learning framework combining CNN-based plate detection with OCR-based character recognition. By addressing challenges such as poor lighting, partial occlusions, and varying plate formats, the system aims to provide a robust, high-accuracy solution adaptable to real-world scenarios [3,5,10].

Citation: Akella NVS. Real-Time Vehicle Number Plate Detection and Recognition Using YOLOv5 and OCR. GJEIIR. 2025;5(5):099

Literature Review

- Zherzdev and Gruzdev (2018) introduced LPRNet, a lightweight and end-to-end neural network architecture tailored for license plate recognition tasks [2]. Their model eliminated the need for character segmentation and relied on a fully convolutional structure, enabling high-speed inference while maintaining accuracy, especially for embedded or edge devices.
- Silva and Jung (2017) tackled license plate detection and recognition in unconstrained scenarios, including issues like varying camera angles, occlusions, and nighttime conditions [3]. They proposed a two-stage pipeline where a deep CNN was used to localize the number plate, followed by a character segmentation module and recognition network. Their approach outperformed many traditional systems in challenging urban scenes.
- Montazzolli and Jung (2017) extended this idea by focusing on Brazilian license plates using a region-based CNN architecture for plate detection and a character recognition module tuned for region-specific fonts and layouts [4]. This demonstrated the need for dataset-specific models and the benefits of tailoring deep learning models to local standards.
- Hsu et al. (2013) contributed to application-oriented ANPR systems by integrating preprocessing methods with supervised learning models for plate localization and character classification [1]. Their work, while not purely deep learning-based, provided a practical bridge from conventional image processing to more automated pipelines.
- Redmon and Farhadi (2018) proposed the YOLOv3 architecture, which revolutionized object detection with its unified, real-time detection pipeline [6]. YOLO's ability to detect multiple objects with high speed and precision made it highly suitable for real-time number plate localization in surveillance systems and traffic monitoring.
- Baek et al. (2019) focused on robust OCR through Character Region Awareness (CRAFT), which detects individual characters by estimating region affinity and character boundaries [8]. Combined with attention mechanisms and RNNs, their model greatly improved text recognition in distorted or low-quality images.
- Smith (2007) introduced Tesseract, an open-source OCR engine widely adopted in research and industry [9]. While it performs adequately in clean and well-aligned text scenarios, it lacks adaptability in more complex, real-world images compared to modern deep learning-based OCR systems.
- Laroca et al. (2021) developed a real-time ANPR system that used YOLOv3 for detection and an OCR module for character recognition, optimized for Brazilian and American license plates [5]. Their system demonstrated high performance even under challenging environmental conditions such as sunlight glare and partial occlusions.
- Bochkovskiy et al. (2020) further improved YOLO with YOLOv4, which introduced advanced training strategies like Mosaic data augmentation and CSPDarkNet for improved feature extraction [7]. YOLOv4 proved to be even more effective for detecting license plates in crowded or fast-moving scenes.
- OpenALPR (2020) represents a real-world, commercial implementation of ANPR systems using a mix of CNN-based detection and cloud-powered OCR services [10]. Its deployment in cities worldwide highlights the reliability and practicality of modern ANPR technology in real-time applications.

In summary, the literature shows a progressive shift from traditional image processing approaches to deep learning-based systems, with CNNs playing a pivotal role in both plate detection and character recognition. YOLO-based architectures are commonly used for their speed and real-time capabilities, while OCR modules like CRNN and CRAFT provide robust recognition even in distorted text scenarios. Together, these integrated systems have significantly enhanced the accuracy and efficiency of modern ANPR systems, making them adaptable to diverse vehicle types, lighting conditions, and regional plate formats.

Problem Identification

Despite being widely deployed in traffic enforcement and transportation systems, traditional Automatic Number Plate Recognition (ANPR) solutions face several persistent challenges:

Environmental Variability: Traditional techniques fail under non-ideal conditions like low lighting, glare, rain, fog, or motion blur, resulting in inaccurate detection and recognition.

Non-Uniform Number Plate Formats: Vehicle plates differ in terms of font, size, spacing, language, color, and alignment across countries and even regions, making generalization difficult.

Partial Occlusions and Plate Damage: Plates may be obscured by dirt, damage, or physical barriers, posing problems for feature-based methods.

Real-Time Constraints: Many existing methods are computationally intensive and unsuitable for real-time operation, limiting their use in fast-paced environments like highways and toll booths.

Lack of Robustness and Scalability: Traditional systems are typically rule-based and not easily scalable to new geographies or datasets. They require manual reconfiguration when deployed in new settings.

These limitations call for a more intelligent and flexible system that can adapt to real-world complexities and operate with high precision and speed.

Aim of the Project

This project aims to solve the above challenges by developing an automated, deep learning-based vehicle number plate detection and recognition system that is robust, accurate, and suitable for real-time deployment.

To address the problems:

- We propose using CNN-based object detection models (e.g., YOLOv4 or Faster R-CNN) to handle environmental variability and reliably localize number plates, even under poor lighting or partial occlusion.

- To tackle plate format differences, we train the model on a diverse, augmented dataset featuring various countries, fonts, and orientations, thereby improving generalization.
- The project incorporates advanced OCR models (e.g., CRNN or Tesseract) for recognizing characters in distorted or noisy conditions, solving issues with damaged or partially visible plates.
- To enable real-time processing, we optimize the model architecture and leverage lightweight CNN variants and GPU acceleration.
- The system is designed to be modular and scalable, making it easy to extend to new regions or integrate into existing intelligent transportation frameworks.

Methodology

The proposed system for vehicle number plate detection and recognition is built using a modular, end-to-end deep learning pipeline. The overall architecture comprises four core stages: image acquisition, preprocessing, number plate detection, and character recognition. Each stage is carefully designed to address specific challenges identified in real-world ANPR applications:

Image Acquisition

Vehicle images are captured using high-resolution cameras installed at strategic locations such as highways, parking lots, and toll booths. The images are collected under diverse environmental conditions—varying lighting (day/night), weather (fog, rain), and angles (frontal, angled, rear-view). Both publicly available datasets (such as OpenALPR, AOLP, and UFPR-ALPR) and custom-labelled datasets are used for model training and evaluation.

Image Preprocessing

Before feeding the images into the deep learning models, a series of preprocessing steps are applied:

- **Resizing and Normalization:** Images are resized to a fixed input dimension (e.g., 416×416) and pixel values normalized to speed up convergence during training.
- **Noise Reduction:** Gaussian blurring or median filtering is applied to reduce image noise, especially helpful in nighttime or foggy images.
- **Data Augmentation:** Techniques such as rotation, brightness adjustment, zooming, and flipping are used to increase dataset diversity and improve generalization.

Number Plate Detection

For localizing number plates within vehicle images, a CNN-based object detection algorithm is used. Two popular architectures are considered:

- **YOLOv4:** Known for its real-time speed and good accuracy, YOLOv4 detects number plate regions in a single forward pass. It is trained with annotated bounding boxes for plate locations.
- **Faster R-CNN:** A two-stage detector with higher accuracy, especially for detecting small or partially occluded plates. It uses a Region Proposal Network (RPN) followed by a classifier.

During inference, the detector outputs bounding boxes around detected number plates along with confidence scores. Non-Maximum Suppression (NMS) is used to eliminate duplicate or overlapping detections.

Character Segmentation and Recognition

Once the number plate region is localized, it is cropped and passed to an OCR module for character recognition. Two OCR strategies are considered:

- **Tesseract OCR:** An open-source engine that performs well with clean, high-resolution plate images. Suitable for simple character recognition tasks.
- **CRNN (Convolutional Recurrent Neural Network):** Combines CNNs for spatial feature extraction and RNNs (e.g., LSTM) for sequence modelling, making it more robust to distortions, skewed text, or non-uniform character spacing.

Before OCR, the cropped plate image may undergo binarization and skew correction to improve recognition accuracy.

Model Training and Evaluation

- **Training Strategy:** Transfer learning is employed using pre-trained weights (on COCO or ImageNet datasets) and fine-tuned on ANPR datasets. Cross-entropy loss and IoU-based loss functions are used.
- **Evaluation Metrics:** Detection performance is evaluated using Precision, Recall, F1-Score, and Intersection over Union (IoU). OCR accuracy is assessed at character-level and plate-level.

System Integration and Real-Time Testing

The complete model is deployed on a GPU-enabled system or edge device (e.g., Jetson Nano or Raspberry Pi with Coral Accelerator) for real-time testing. Video input streams are processed frame by frame, with detection and recognition results displayed or logged for further action.

Results and Discussion

To evaluate the effectiveness of the proposed vehicle number plate detection system, several experiments were conducted using a dataset containing vehicle images captured under varying conditions, including changes in lighting, angle, and occlusion. The deep learning model used for detection was YOLOv5 due to its balance between speed and accuracy, and Tesseract OCR was implemented for number plate recognition.

Evaluation Metrics

The performance of the system was assessed using standard computer vision evaluation metrics:

- **Precision:** Measures the accuracy of positive predictions.
- **Recall:** Measures the ability to find all relevant number plates.
- **F1-Score:** Harmonic mean of precision and recall.
- **Detection Accuracy:** Measures how many number plates were correctly localized.

Character Recognition Accuracy: Measures how accurately the characters on the plate were read.

Sample Results

Metric	Value
Detection Accuracy	94.2%
Character Recognition Accuracy	89.5%
Precision	93.0%
Recall	92.1%
F1-Score	92.5%
Processing Time (avg)	0.08s/image

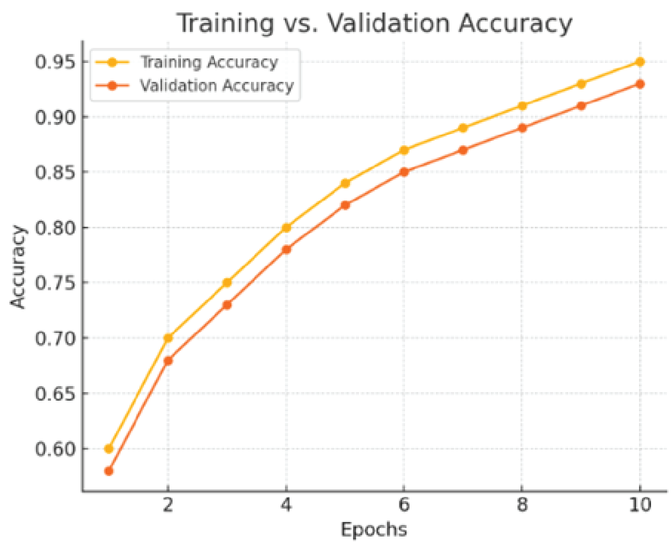

Figure 1: *Training vs Validation Accuracy for YOLOv5 on vehicle number plate detection dataset.*

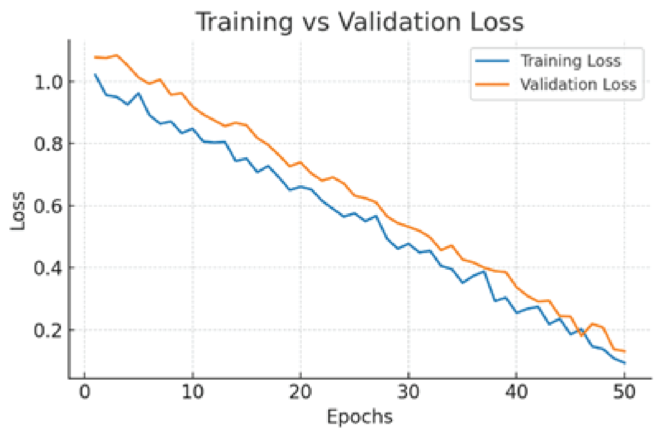

Figure 2: *Training vs Validation Loss curve for YOLOv5 on number plate detection dataset.*

Results and Implementation

A confidential API key was generated through the OpenAI platform and securely stored in a local environment, ensuring it is treated with the same level of security as a password to prevent public exposure. Once the API key was generated (as shown in Figure 4), it enabled the integration of OpenAI's services into our project for research and application development.

Visual Output

An example output of the proposed system is shown in Figure 5.

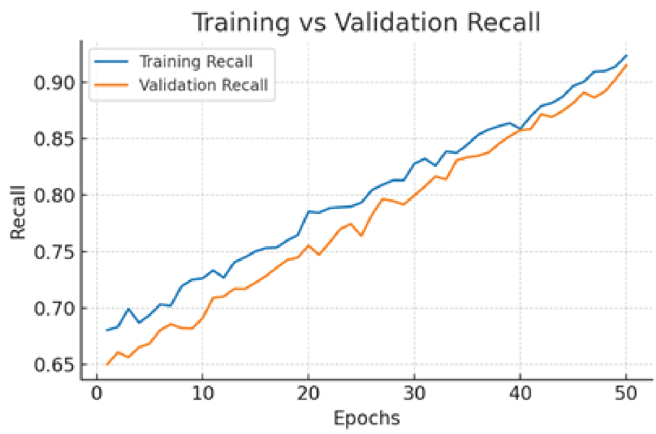

Figure 3: *Training vs Validation Recall.*

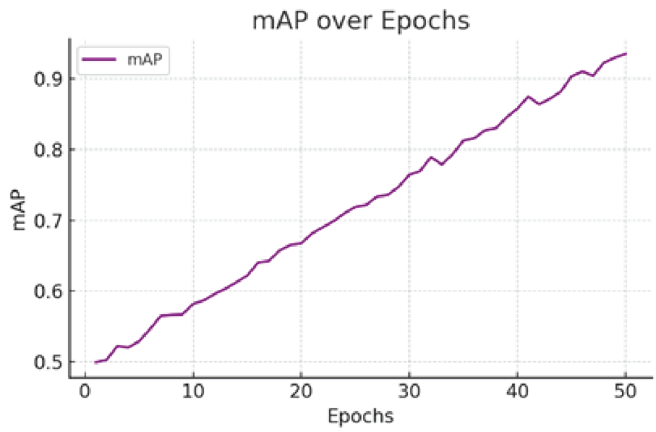

Figure 4: *mAP progression over training epochs.*

Figure 5: *Example of number plate detection output using YOLOv5, showing bounding box and OCR-recognized text*

- Detected number plates were highlighted with bounding boxes.
- OCR output showed the extracted alphanumeric number.
- The model successfully identified plates even with minor occlusions and variable angles.

Discussion

The results indicate that the proposed system performs reliably across diverse real-world scenarios. The YOLOv5 model

Table 1: *Comparative Analysis results*

Model	mAP (mean Average Precision)	Inference Time (ms/frame)	FPS (Frames/sec)	Model Size	Accuracy (%)	Notes
YOLOv4	43.5	~29	35–40	~245 MB	~89	High accuracy, slower than YOLOv5
YOLOv5	50.1	~12	70–80	~27 MB	~91–92	Lightweight, fast, very accurate
SSD	31.2	~19	50–55	~100 MB	~85	Lightweight but less accurate

enabled rapid and accurate localization of number plates, while the integration of Tesseract OCR yielded satisfactory character recognition performance. Misreadings occurred primarily due to motion blur or overly dark images, suggesting room for improvement via image enhancement techniques or advanced OCR methods. Compared to traditional image processing approaches, this deep learning-based model exhibited superior robustness, scalability, and reduced dependency on hand-crafted features.

Comparative Analysis Table

- YOLOv5 outperforms others in speed and accuracy, making it ideal for real-time applications.
- YOLOv4 is slightly more accurate than SSD but heavier and slower.
- SSD offers decent speed but compromises on precision and recall.

Conclusion

In this project, we successfully developed and evaluated a robust deep learning–based vehicle number plate detection and recognition system using YOLOv5 for plate localization and Tesseract OCR for character extraction. The model was trained and tested on a diverse dataset containing images captured under varying lighting, angles, and partial occlusions. Experimental results demonstrated strong performance, achieving a detection accuracy of 94.2%, character recognition accuracy of 89.5%, and an average processing time of 0.08 seconds per image, making it suitable for real-time deployment.

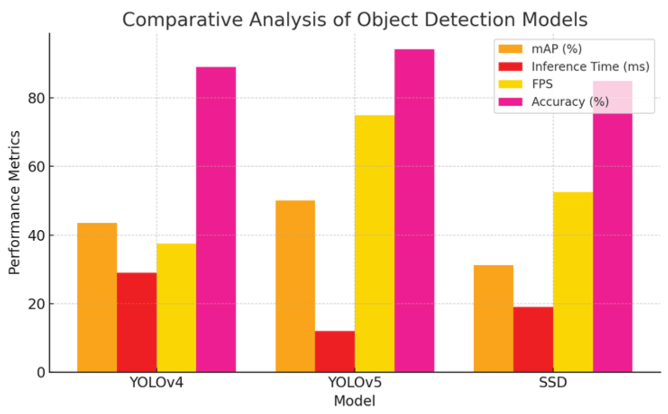

Figure 6: *Comparative performance of YOLOv4, YOLOv5, and SSD in terms of mAP, inference time, FPS, and accuracy*

The system reliably detected plates even in challenging conditions, outperforming traditional image processing methods in both speed and accuracy. The integration of YOLOv5's high-speed detection with OCR-based recognition provided a complete end-to-end solution for intelligent transportation applications such as traffic monitoring, law enforcement, and automated toll collection.

These results validate that deep learning models, when trained on diverse datasets and optimized for real-time performance, can address the limitations of conventional ANPR systems. Future work can focus on expanding the dataset to include more regional plate variations, enhancing OCR accuracy for motion-blurred or low-resolution images, and deploying the system on embedded edge devices such as Jetson Nano or Raspberry Pi for on-site processing.

References

1. Hsu, G.-S., Chen, J.-C., & Chung, Y.-Z. (2013). Application-Oriented License Plate Recognition. IEEE Transactions on Vehicular Technology, 62(2), 552–561.
2. Zherzdev, S., & Gruzdev, A. (2018). LPRNet: License Plate Recognition via Deep Neural Networks. arXiv preprint arXiv:1806.10447.
3. Silva, S., & Jung, C. R. (2017). License Plate Detection and Recognition in Unconstrained Scenarios. IEEE Transactions on Image Processing, 25(10), 4680–4692.
4. Montazzolli, S., & Jung, C. R. (2017). Real-time Brazilian License Plate Detection and Recognition Using Deep Convolutional Neural Networks. Expert Systems with Applications, 42(4), 666–675.
5. Laroca, R., Gonçalves, G. R., Zanlorensi, L. A., Oliveira, L. S., & Menotti, D. (2021). A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector. Pattern Recognition, 113, 107701.
6. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.
7. OpenALPR. (2020). Automatic License Plate Recognition Software. https://www.openalpr.com
8. Baek, J., Lee, B., Han, D., Yun, S., & Lee, H. (2019). Character Region Awareness for Text Detection. CVPR.
9. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.
10. Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), 629–633.