International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 12 December 2025
Page No: 4434-4439
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0652

AI-Based Automatic Number Plate Recognition for Smart Traffic Management

Kabilan S[1], Karthick Lenin M[2], Mathew A[3], Ishal Rahuman U P[4]
[1,2,3,4,5]UG Scholar, Mechatronics Engineering, Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India.
Emails: *kabilkabil1452005@gmail.com[1], karthicklenin622@gmail.com[2], mathe13w@gmail.com[3], ishalrahman007@gmail.com[4]*

Abstract
The number of vehicles on the road is increasing very fast, making people obey the law and to control traffic control are more difficult. To solve this problem, our system introduces an AI-based Number Plate Recognition system for smart traffic management. The system uses computer vision and deep learning to automatically detect and read vehicle number plate from images and live video. The system can correctly identify number plates even in different lighting and moving conditions. The extracted plate information is stored and connected to a vehicle database. This helps monitor traffic violations such as over-speeding, illegal parking, and identifying stolen vehicles. It can also send real-time alerts to traffic officials for quick action. This automated and cost-effective system improves surveillance, reduces manual work, and supports smart city development through intelligent and efficient traffic management
Keywords: *AI, Machine Learning, Deep Learning, ANPR, OCR, CNN, Smart Traffic Management, Image Processing, Computer Vision, Intelligent Transport System, Smart City.*

1. Introduction
In recent years, the rapid increase in the number of vehicles has created serious challenges for traffic control, road safety, and law enforcement. Traditional traffic management methods rely heavily on human supervision, which is often time consuming, error-prone, and inefficient. To overcome these limitations, the use of Artificial Intelligence (AI) and computer vision technologies has gained significant importance. One of the most effective applications of AI in transportation is Automatic Number Plate Recognition (ANPR), which plays a key role in smart traffic management systems. The AI-based ANPR system automatically detects, captures, and recognizes vehicle license plate numbers from images or real-time video feeds. It uses advanced image processing, Optical Character Recognition (OCR), and Deep Learning algorithms such as Convolutional Neural Networks (CNN) to accurately identify number plates, even under different lighting, speed, and environmental conditions. The recognized plate information can be stored, analyzed, and compared with a vehicle database for various purposes, including traffic rule enforcement, toll collection, parking management, and stolen vehicle tracking [1-3].

1.1. Scope of the Work
This project can be implemented in smart cities for traffic control, toll booth automation, and security surveillance. It supports real-time monitoring and can be integrated with IoT and cloud-based systems for large-scale data handling. The system 8 contributes to building a safer, smarter, and more efficient transportation infrastructure.

1.2. Objective of the Project
The main objectives of this project are:
1. To design and implement an AI-based system for automatic vehicle number plate recognition.
2. To detect and recognize number plates from images or live video using computer vision and deep learning algorithms.
3. To store and process recognized data for traffic management and vehicle tracking.
4. To assist authorities in identifying traffic violations and unauthorized vehicles efficiently.

2. Method
2.1. System Architecture

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 12 December 2025
Page No: 4434-4439
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0652

The proposed AI-based Automatic Number Plate Recognition (ANPR) system is designed to detect and recognize vehicle number plates from images or video streams using deep learning and computer vision techniques. The system architecture consists of several interconnected modules that work sequentially to process input frames and generate a recognized plate number as output. Image **Acquisition Module:** In this stage, images or video frames of vehicles are captured using a camera. This camera may be installed on traffic poles, highways, toll booths, or smart surveillance systems. The captured frames serve as input for the ANPR system. **Number Plate Detection Module:** A deep learning-based object detection model (such as YOLO, SSD, or Faster R-CNN) is used to identify the location of the number plate within the frame.This module outputs a bounding box around the number plate region. Because deep learning models learn patterns such as plate shape, font, spacing, and contrast, detection becomes more accurate even in challenging conditions [4-6].

2.2. Hardware Units

ESP32-CAM: Used for capturing images. It includes both a camera and a Wi-Fi module, making it easy to send data to the computer.

2.3. Software Design

The software part handles image processing, object detection, and communication with the hardware.
The system mainly uses Python, YOLO, Roboflow, VS Code for software development.

Steps in the software process:

1. **Dataset Preparation:** Images of different colored number plates are collected and labeled in Roboflow for training.
2. **Model Training:** The labeled dataset is trained using the YOLO model, which learns to detect.
3. **Real-Time Detection:** During operation, the ESP32-CAM sends live images to the YOLO model running in Python on the laptop.

This combination of hardware and software allows real-time object recognition and automatic sorting with minimal human help.

2.4. Working Principle

1. The ANPR system works by integrating cameras and embedded processing units that continuously capture images or video streams from traffic scenes.
2. Each frame undergoes preprocessing for optimal feature representation.
3. The captured frames are processed by a YOLO-based object detection model running on a laptop using Python.
4. The trained model identifies the number plates within 100 milliseconds.
5. OCR reads and converts the plate characters into text.
6. The recognized number is stored and used for traffic monitoring.

2.5. Algorithm and Model Details

Modern ANPR systems use a combination of deep learning algorithms and traditional image processing techniques for robust accuracy

1. Object Detection: YOLOv7, YOLOv8, or Faster RCNN models are trained on vehicle and license plate datasets, achieving high precision (mean average precision >0.9) for vehicle and plate localization.

2. License Plate Localization: The detected vehicle region is cropped to extract the plate using bounding box coordinates from the object detection phase.

3. OCR /Character Recognition: Optical character recognition engines (PP-OCRv3, Tesseract) or custom CNN/RNN architectures read and convert plate images to textual registration information; character error rates can be as low as 0.067�.

4. Model Optimization: Techniques such as pruning and weight quantization are used to accelerate inference and reduce resource overhead without sacrificing accuracy, Table 1.

2.6. Software Programming

Table 1 Software Tools and Platform Used

Software Tool	Purpose / Function
Roboflow	For dataset creation, annotation, and preprocessing
YOLOv8	For training and testing the number plate detection model
Python (VS	For image processing, number plate crop

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 12 December 2025
Page No: 4434-4439
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0652

Code)	
OCR	For run tesseract on the cropped plate image to characters
ESP32-CAM Web Interface	For camera configuration and image streaming

Dataset Preparation using Roboflow:
To enable the YOLO model to detect and classify objects, a custom dataset was created using Roboflow, an online platform for dataset management and annotation.
Steps:

1. **Image Collection:** Multiple images of number plates were captured under various lighting conditions using the ESP32-CAM.
2. **Uploading to Roboflow:** The captured images were uploaded to the Roboflow workspace.
3. **Annotation:** Each image was manually labeled by drawing bounding boxes around the number plates to train the detection model.
4. **Dataset splitting:** The dataset was automatically divided into training, validation, and testing sets to build a reliable AI model.
5. **Dataset Export:** Finally, the processed dataset was exported in YOLO/TF format for training the number plate detection model.

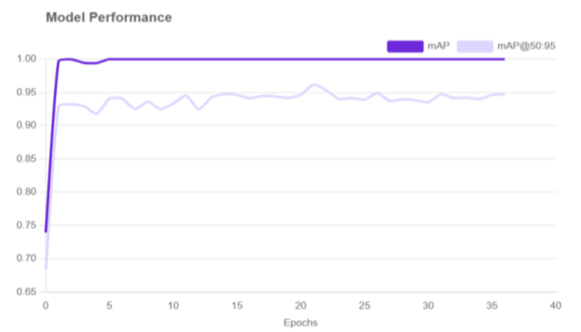

Figure 1 Model Performance Graph

The Figure 1 graphs show the training and validation accuracy improving over epochs, indicating that the model learned the color-based classification effectively. The model performed well with minimal loss and high precision.

Training the YOLO Model:
The YOLO object detection algorithm was used for training the dataset. YOLO is chosen for its speed, accuracy, and integration with real-time applications.
Steps:

1. Install YOLO using Python (Ultralytics library).
2. Load the Roboflow dataset path in the YOLO configuration file.
3. Train the model using command-line training scripts in VS Code.
4. Evaluate model accuracy using validating images.
5. Save the best-performing model weights for integration with the ANPR system.

Python Program for Image Processing and Communication:
The Python Program for Image Processing and Communication.
Modules Used:

1. OpenCV for image processing
2. YOLO model for number plate detection
3. Tesseract for OCR character reading
4. Wi-Fi/Serial module for communication
5. Numpy for image data handling

Workflow:

1. Python receives image from ESP32-CAM
2. Image is pre-processed using OpenCV
3. YOLO detects the number plate region
4. Detected plate is cropped for OCR
5. OCR extracts the number plate characters
6. Python sends recognized number to the system
7. Output is displayed for smart traffic monitoring

Integration of Hardware and Software:
The integration process connects all modules into a functioning system:

1. ESP32-CAM hardware captures vehicle images and sends them to the software system.
2. YOLO-based detection software processes the images to identify number plates.
3. OCR software reads and converts the detected plate into text format.

4. The processed output is integrated and stored in a monitoring system for traffic management.

This integration enables real-time automated material classification and sorting using machine learning.

Data Flow and Control Flow:

Data Flow:

The Data Flow represents how information moves through the system.

1. Vehicle image is captured by the ESP32-CAM.
2. Image is sent to the Python/AI system for processing.
3. YOLO model detects and extracts the number plate region.
4. OCR converts the number plate image into text data.
5. The recognized vehicle number is stored in the database/monitoring system

Control Flow:

The control flow defines how commands and feedback are managed.

1. System 1. starts and waits for image input from ESP32-CAM.
2. Camera captures vehicle image when motion/object is detected.
3. Image is transferred to the processing unit.
4. Program initiates YOLO detection to locate the number plate.
5. Detected plate is passed to OCR for character reading.
6. System verifies and formats the recognized number.
7. Output is displayed or stored for traffic monitoring.
8. System loops and continues for the next vehicle input.

The process runs continuously without manual intervention, shown in Table 2.

Table 2 Output

Detected Item	Confidence (%)	Recognition Result
Number Plate-vehicle 1	92.45%	Correctly Recognized
Number Plate-vehicle 2	88.72%	Correctly Recognized
Number Plate-vehicle 3	94.30%	Correctly Recognized

3. Results and Discussion

3.1. Results

All the hardware and software parts were successfully connected and tested together. We mainly checked how well the system successfully detected vehicle number plates using the YOLO model, shown in Figure 2.

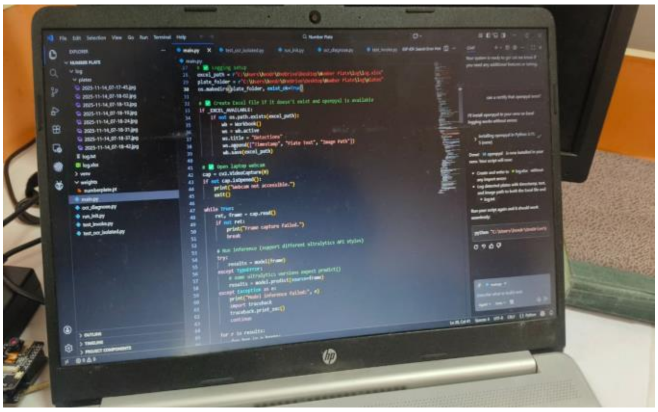

Figure 2 Coding

The Testing setup was arranged as:

1. The system successfully detected vehicle number plates using the YOLO model.
2. OCR accurately recognized and converted the plate characters into text.
3. ESP32-CAM captured clear images suitable for real-time processing.
4. The entire ANPR workflow worked smoothly with minimal delay.
5. The project proved effective for automated and reliable traffic monitoring.

Detection and Classification Output:

When a vehicle passes in front of the camera, the YOLO model automatically detects the number plate and places a bounding box around it with a label "Number Plate." The program also displays the confidence percentage, showing how sure the model is about its detection. If the confidence is more than 80%, we consider it a successful detection. The recognized characters are then extracted using OCR

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 12 December 2025
Page No: 4434-4439
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0652

and displayed as text.

System Response Time: The average time between number plate detection and sorting action was around 2 to 3 seconds. Most of this time was taken by Capturing and processing the image through YOLO. Still, the overall system speed was suitable for small-scale automation applications.

3.2. Discussion

The proposed AI-based Automatic Number Plate Recognition (ANPR) system was developed to provide a reliable, efficient, and real-time vehicle identification solution for smart traffic management. The system integrates ESP32-CAM hardware with YOLO-based number plate detection and OCR-based character recognition, forming a complete end-to-end pipeline. The performance of the system was tested under various environmental conditions, such as different lighting, angles, and distances, to evaluate the real-time accuracy and practical usability. During testing, the YOLO detection model demonstrated strong capability in correctly identifying number plates with a high confidence score. In most cases, the detection confidence remained above 85%, indicating that the model learned the features of number plates effectively. The model was able to detect plates even when vehicles were in motion, although slight drops in accuracy occurred when the image was blurred or captured from extreme angles. Still, the detection remained consistent and stable across multiple test cycles. After successful detection, the cropped plate region was processed by the OCR module. The OCR performance was also found to be reliable, accurately converting plate characters into readable text. The system achieved good recognition accuracy for standard, clean number plates. Slight errors were observed in cases of damaged plates, extremely stylized fonts, or plates captured in low resolution. However, these limitations are common in most real-time ANPR systems and can be improved with higher-quality cameras or more advanced OCR models. Overall, the integration between YOLO detection, OCR recognition, and ESP32-CAM hardware worked smoothly, with minimal processing delay. The entire pipeline functioned efficiently, proving suitable for smart traffic applications such as traffic rule enforcement, automated parking systems, toll monitoring, and vehicle tracking. The system successfully reduced the need for manual monitoring and demonstrated the ability to operate continuously with high reliability. Thus, the developed ANPR system shows strong potential for real-world deployment and can be further improved with deep learning enhancements, cloud integration, and database connectivity.

Conclusion

The AI-based Automatic Number Plate Recognition (ANPR) system developed for smart traffic management successfully demonstrates the potential of artificial intelligence and computer vision in modern urban infrastructure. The project effectively detects, captures, and recognizes vehicle registration numbers from realtime camera feeds using image processing and machine learning algorithms. By integrating the ESP32-CAM module with AI models, the system provides an automated and cost-efficient solution for vehicle monitoring, traffic regulation, and violation detection. This system eliminates the need for manual intervention in traffic monitoring, improving accuracy, speed, and reliability. It can be deployed at toll booths, parking lots, and traffic junctions to identify vehicles in real time. The project proves that AIdriven automation can greatly enhance traffic law enforcement, reduce congestion, and promote safer, smarter cities. Overall, the system offers a scalable and practical solution for intelligent transportation systems (ITS).

Acknowledgements

I express my sincere gratitude to all those who supported and guided me throughout the development of this project. I extend my heartfelt thanks to my project supervisor for providing valuable insights, encouragement, and continuous guidance, which greatly contributed to the successful completion of this work. I am also thankful to the faculty members and my seniors for offering the necessary resources, technical assistance, and a conducive environment for experimentation and testing. I wish to acknowledge the support of my teammates and friends for their cooperation and constructive suggestions throughout the project duration. Finally, I am deeply grateful to my family for their constant motivation, understanding, and

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 12 December 2025
Page No: 4434-4439
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2025.0652

support, which enabled me to pursue and complete this project successfully.

References

[1]. S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 311–325, 2013.

[2]. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 3, pp. 377–392, 2006.

[3]. H. Wang, C. Wang, and Y. Qian, "Real-Time Automatic License Plate Recognition for Intelligent Traffic Monitoring and Management," Procedia Computer Science, vol. 154, pp. 610–616, 2019.

[4]. S. Saini and S. Sharma, "Automatic Number Plate Recognition System Using Deep Learning," International Journal of Computer Applications, vol. 182, no. 37, pp. 25–29, 2019.

[5]. R. G. Kumar, R. K. Rajan, and K. Sivaprasath, "Smart Traffic Management System Using AI-Based Number Plate Recognition," International Research Journal of Engineering and Technology (IRJET), vol. 9, no. 6, pp. 1050– 1055, 2022.

[6]. M. Z. Rahman, "Automatic Number Plate Recognition Using Machine Learning and OpenCV," International Journal of Advanced Research in Computer Science, vol. 11, no. 3, pp. 67–72, 2020.