

Vehicle Number Plate Detection System

V. MEGHANA¹, G. MANIDEEP², S. LAKSHMAN³, J. DEVI⁴, Mrs. V. SURYAKALA⁵

¹²³⁴⁵Department of Electronics & Communication Engineering
Sanketika Institute of Technology and Management, Visakhapatnam, AP

Abstract - Vehicle number plate detection is a crucial technology in modern transportation systems, enabling automated identification and tracking for applications such as traffic management and law enforcement. This paper presents the design and implementation of an IoT-based system using a **Raspberry Pi**, **OpenCV**, and **Tesseract OCR**. The system captures images, identifies the number plate region through contour detection, and extracts characters using Optical Character Recognition (OCR). The data is transmitted to a cloud platform for real-time monitoring and mobile alerts. Results demonstrate that the system provides a cost-effective and scalable solution for intelligent transportation infrastructures.

Key Words: Vehicle Number Plate Detection, Raspberry Pi, Internet of Things (IoT), OpenCV, Tesseract OCR, Python

1. INTRODUCTION

Automatic Number Plate Recognition (ANPR) systems have emerged as a vital solution to the errors inherent in manual vehicle identification. While traditional methods are labor-intensive, existing automated systems are often expensive and require specialized infrastructure. The integration of **Internet of Things (IoT)** and edge computing platforms like the Raspberry Pi offers a low-cost, intelligent alternative. This project focuses on a system that processes images locally using image processing algorithms and provides immediate alerts to authorized personnel via cloud connectivity.

2. SYSTEM ARCHITECTURE AND DESIGN

The system architecture follows a linear pipeline: Camera Module → Raspberry Pi → Image Processing → OCR Engine → Cloud Platform.

2.1 HARDWARE IMPLEMENTATION

- **Raspberry Pi:** Acts as the central controller, managing image data and GPIO-linked peripherals.
- **Camera Module:** Captures high-resolution images via the CSI (Camera Serial Interface) port.
- **Alert Module:** Employs a Green LED for successful detection (GPIO17) and a Buzzer for audible notification (GPIO22).

2.2 SOFTWARE DESIGN AND METHODOLOGY

The software is developed in **Python**, utilizing the **OpenCV** library for computer vision tasks.

1. **Preprocessing:** Images are converted to grayscale and filtered using Gaussian techniques to reduce noise.
2. **Edge and Contour Detection:** Canny edge detection and contour algorithms identify rectangular regions matching license plate dimensions.
3. **OCR Character Extraction:** The identified region is passed to the **Tesseract OCR** engine to convert pixel data into digital text.

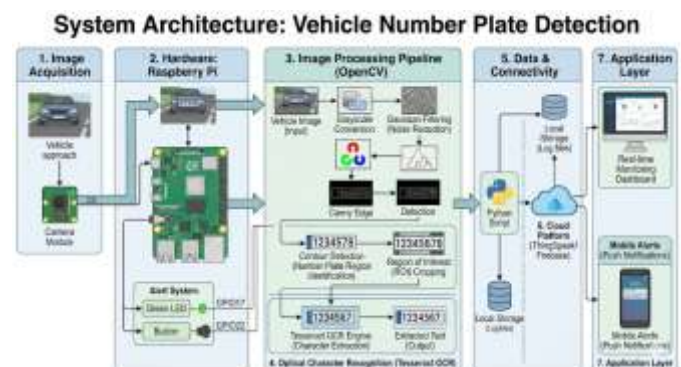


Fig-1: system architecture

3. RESULTS AND TESTING

The system was evaluated under various environmental conditions.

Test Case	Condition	Observation
Case 1	Clear Visibility	High accuracy in detection and extraction.
Case 2	Moving Vehicle	Successful detection with slightly higher processing lag.
Case 3	Low Lighting	Reduced OCR accuracy due to image noise.

Table-1: Test cases

Processing time averaged a few seconds, which is acceptable for smart parking and security checkpoints

4. CONCLUSIONS

The developed system demonstrates the feasibility of low-cost, IoT-enabled vehicle identification. By automating the recognition process, it reduces human effort and increases operational efficiency. While limitations exist in extreme lighting, the architecture provides a foundation for future enhancements, such as integrating **Deep Learning** models like Convolutional Neural Networks (CNNs) for higher accuracy.

ACKNOWLEDGEMENT

The authors express their gratitude to project guide **Mrs. V. Suryakala, M.Tech**, and Dr. R. Suneetha, Ph.D, Head of the ECE Department, for their invaluable technical annotations and facilities

REFERENCES

1. Gonzalez, R. C., Woods, R. E.: Digital Image Processing. Pearson Education.
2. Rosebrock, A.: Practical Python and OpenCV. PyImageSearch Publications.
3. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer.

4. Baldonado, M., et al.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121.