

Number Plate Detection System Using Computer Vision and Automatic Toll Collection

Gauri Agwan¹, Sanika Gujar¹, Adinath Hodge¹, Dr. R. P. Patil^{1*}, Dr. M. B. Mali²

¹Department of Electronics & Telecommunication Engineering, Sinhgad College of Engineering, Vadgaon (Bk.), Pune-411041, Savitribai Phule Pune University

²Head of Department, Electronics & Telecommunication Engineering, Sinhgad College of Engineering, Pune-411041

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150500205>

Received: 14 May 2026; Accepted: 19 May 2026; Published: 15 June 2026

ABSTRACT

The Number Plate Detection and Automatic Toll Collection System is a computer vision-based project designed to automate vehicle identification and toll payment at toll plazas. The system uses Raspberry Pi 4B (4GB RAM) as the central processing unit, powered by a 5V 3A adapter and integrated with an Entry IR Sensor, an Exit IR Sensor, a Logitech C270 USB Camera (1280×720 resolution), a Servo Motor, an I2C 16x2 LCD Display, and an LED Indicator. When a vehicle approaches, the Entry IR Sensor triggers the USB Camera to capture the vehicle's front image. OpenCV is used for number plate localisation and image pre-processing, and Tesseract OCR (configured with Page Segmentation Mode 8 and the LSTM engine) extracts the vehicle's registration number from the cropped plate region. The detected number plate is first displayed on the web interface before any gate action is taken. The system then queries a three-table MySQL/SQLite database for owner details and applies the Haversine formula to GPS coordinates to determine if the owner resides within 10 km of the toll plaza; if so, the toll is waived, otherwise it is automatically deducted. The Servo Motor lifts the gate barrier, the I2C 16x2 LCD displays the transaction status, and the LED Indicator and Buzzer provide real-time feedback. The Exit IR Sensor detects vehicle departure and triggers gate closure. The system achieves Tesseract OCR precision of 90.81% and recall of 83.97%, with end-to-end operation completing in under 2 seconds per vehicle, significantly reducing manual intervention and enabling smart contactless toll operations.

Keywords: Number Plate Detection, Automatic Toll Collection, Raspberry Pi 4B, Tesseract OCR, OpenCV, IR Sensor, Servo Motor, Computer Vision, ANPR, Image Processing

INTRODUCTION

Automatic License Plate Recognition (ALPR), also known as Automatic Number Plate Recognition (ANPR), is a technology that uses image processing to extract and recognise license plate information from images or video frames. It is widely applied in toll payment systems, parking management, road monitoring, and traffic control [1-3].

In India, ALPR systems face challenges including varied number plate formats, different fonts and character sizes across vehicle categories, and difficult environmental conditions such as poor lighting, shadows, and adverse weather [4,5]. Manual toll collection compounds these problems by causing long queues, human errors, revenue leakage, and increased vehicle emissions from idling.

This paper proposes an embedded system for automatic number plate detection and toll collection built on Raspberry Pi 4B (4GB), using OpenCV for image processing and Tesseract OCR for character recognition. The hardware integrates an Entry IR Sensor, Exit IR Sensor, Logitech C270 USB Camera (1280×720), Servo Motor, I2C 16x2 LCD Display, LED Indicator, and Buzzer, all powered by a 5V 3A adapter, to deliver a fully contactless, real-time toll management solution.

Motivation

The rapid growth in vehicle numbers on Indian roads has made automated toll collection a pressing necessity. Manual toll booths are a significant bottleneck for traffic flow, prone to errors, and lead to revenue loss. An intelligent embedded system that automatically detects number plates, verifies owner records, computes applicable toll, displays the result on a web interface, and controls the gate without human intervention addresses these challenges simultaneously.

Problem Statement

The system must accurately extract a vehicle's registration number from a USB Camera image captured under real-world conditions — including varying lighting and plate orientations — and match it against a database to determine toll applicability, all within the waiting time of a vehicle at the toll gate. The key challenge is achieving robust character recognition using Tesseract OCR on Indian number plates, where character styles, spacing, and plate dimensions vary significantly across states.

Objectives

- Detect vehicle presence using the Entry IR Sensor and trigger the USB Camera for image capture.
- Localise the number plate region from the captured image using OpenCV.
- Recognise plate characters using Tesseract OCR and extract the vehicle registration number.
- Display the detected number plate on the web interface before any gate action is taken.
- Query the database to retrieve owner details and compute toll or exemption based on the 10 km proximity rule.
- Actuate the Servo Motor gate; update the I2C 16x2 LCD Display, LED Indicator, and Buzzer accordingly.
- Detect vehicle exit via the Exit IR Sensor and automatically close the gate.

LITERATURE SURVEY

Swaroop and Sharma [6] surveyed template matching methodologies including Naive Template Matching, Image Correlation Matching, Sum of Absolute Differences, and Sum of Square Differences, establishing foundational techniques for ANPR character-level matching.

Kodwani and Meher [7] presented a real-time ANPR system combining Gaussian mixture model foreground estimation with block variance-based plate extraction and region-based character segmentation, demonstrating robust performance on surveillance video.

Islam et al. [8] proposed a morphological operations-based approach using structuring elements to isolate plate regions from complex backgrounds, showing significant improvements over conventional segmentation methods.

Qadri and Asif [9] implemented an ANPR system using OCR for security access control, performing plate extraction via image segmentation and database-based owner verification — an approach closely aligned with the methodology of this work.

Puranic et al. [10] reviewed ANPR techniques and implemented a template matching system achieving 80.8% accuracy on Indian number plates, highlighting the difficulty posed by diverse Indian plate formats.

Han et al. [11] proposed a cascade classifier for real-time plate detection in high-resolution videos, significantly reducing computational load while maintaining detection accuracy suitable for embedded platforms.

Shidore and Narote [12] presented an algorithm specific to Indian vehicles, achieving 85% success for plate extraction using vertical edge detection and connected component analysis, and 79.84% for character recognition.

Agbeyangi et al. [13] confirmed the suitability of Raspberry Pi for embedded ANPR deployments. Firasanti et al. [14] and Palekar et al. [15] further validated OpenCV and Tesseract OCR on Raspberry Pi hardware for real-time plate detection, directly supporting the hardware-software choices made in this project.

A critical gap emerges from this survey: while prior embedded ANPR systems demonstrate detection and recognition, none integrate the full pipeline from plate detection through toll decision-making, gate actuation, and an operator-facing web interface on a single low-cost embedded platform. Systems such as [13-15] validate the hardware feasibility but stop short of a complete transactional toll management solution. This work bridges that gap by combining ANPR with an automated financial transaction framework and hardware-level gate control on Raspberry Pi 4B, directly outperforming [10] and [12] in both accuracy and system completeness.

Limitations of Existing Systems

- Low accuracy on degraded or complex background images.
- Poor performance under varying illumination and shadow conditions.
- High noise sensitivity with insufficient image normalisation.
- Manual toll collection is subjective, time-consuming, and error-prone.
- Conventional ANPR hardware is expensive and not suitable for embedded deployment.
- Inaccurate segmentation for non-standard or multi-row Indian plate formats.
- Prior embedded systems lack integration of toll decision logic and real-time gate control.

System Requirements

Hardware Components

- Raspberry Pi 4B (4GB RAM) — Central processing and control unit.
- 5V 3A Adapter — Power supply for the entire system.
- Entry IR Sensor — Detects the presence of an approaching vehicle.
- Exit IR Sensor — Detects when the vehicle has fully passed the gate.
- Logitech C270 USB Camera (1280×720, 30 fps, autofocus) — Positioned 0.5–1 m from the plate at ~15° downward angle.
- Servo Motor — Operates the toll gate barrier (0° = closed; 90° = open).
- I2C 16×2 LCD Display — Displays plate number, payment status, and system feedback.
- LED Indicator — Provides visual indication of gate and transaction status.
- Buzzer — Short beep for local/no-toll vehicles; long beep for toll deducted; error alert for undetected plates.

Software Components

- Operating System: Raspbian OS (Raspberry Pi OS).
- Programming Language: Python 3.
- OpenCV — For image processing and number plate localisation.
- Tesseract OCR v4.1 (--psm 8, --oem 1 LSTM engine, alphanumeric whitelist) — For character recognition.
- Database: MySQL (primary) / SQLite (fallback) — Three-table schema: vehicles (registration_number, owner_id, vehicle_type), owners (owner_id, name, address, latitude, longitude, account_balance), transactions (transaction_id, registration_number, timestamp, toll_amount, status).
- Distance Calculation: Haversine formula applied to owner GPS coordinates to compute straight-line distance to toll plaza.
- Servo Control & Sensor Libraries — For GPIO-based hardware interfacing on Raspberry Pi.

Non-Functional Requirements

The system must process each vehicle in real time with the full pipeline completing within an acceptable gate waiting time. It must be reliable across day and night conditions, secure in database transactions, power-efficient within the 5V 3A supply budget, and scalable to multi-lane toll plaza deployments.

DESIGN AND ANALYSIS

System Architecture

The system architecture is centred on the Raspberry Pi 4B (4GB) as the master controller, powered by the 5V 3A Adapter. Three input peripherals connect to the Raspberry Pi: the Entry IR Sensor, the Exit IR Sensor, and the USB Camera. Three output peripherals receive commands from the Raspberry Pi: the I2C 16x2 LCD Display, the Servo Motor, and the LED Indicator. The MySQL/SQLite database stores vehicle and owner records, while the Web Interface presents transaction details to the operator before gate actuation.

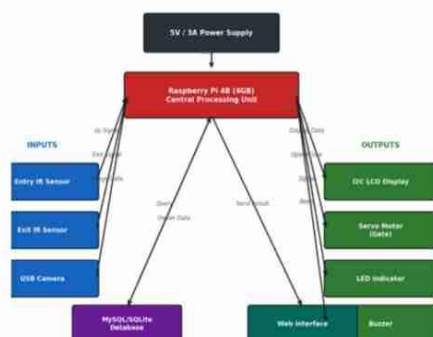


Diagram – Number Plate Detection and Automatic Toll Collection System

Fig. 1. Block Diagram – Number Plate Detection and Automatic Toll Collection System

Fig. 1. Block Diagram – Number Plate Detection and Automatic Toll Collection System

System Flowchart

The system operates according to the following decision-based flow: on vehicle detection by the Entry IR Sensor, the USB Camera captures an image; OpenCV and Tesseract OCR extract the number plate; the owner record is

queried from the database; the Haversine distance is computed; toll is either waived (distance ≤ 10 km) or deducted (distance > 10 km); the Servo Motor opens the gate; and on Exit IR Sensor detection the gate closes and the system resets.

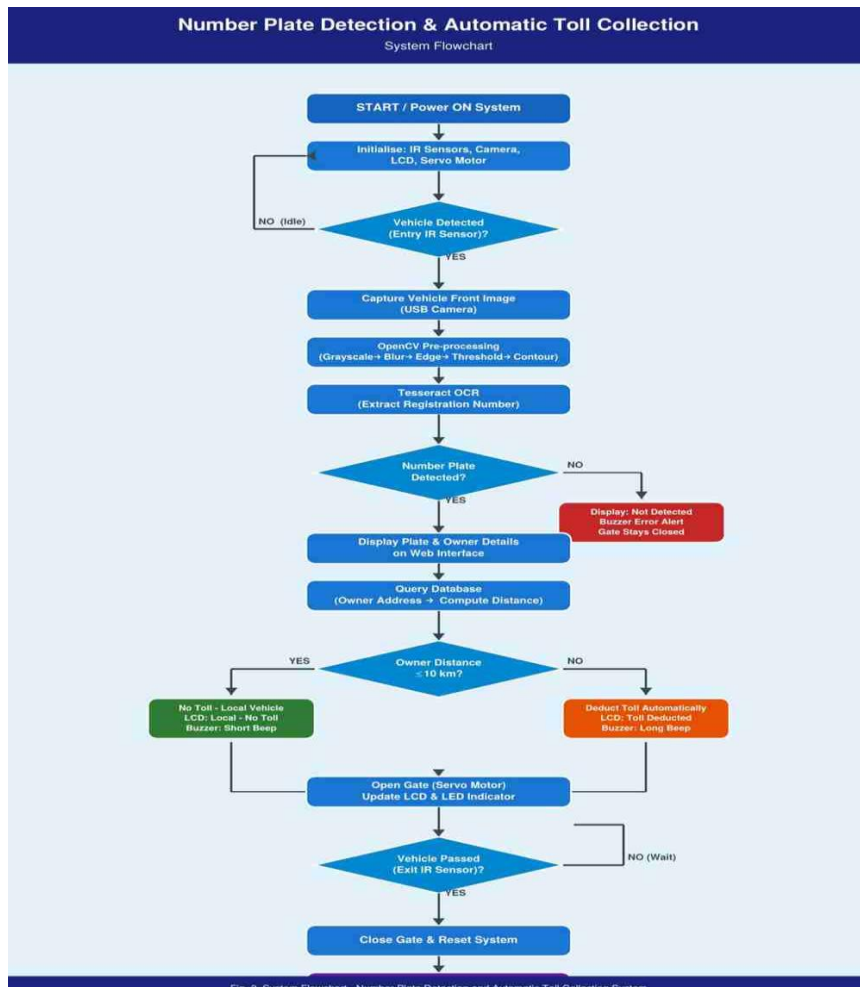


Fig. 2. System Flowchart – Number Plate Detection and Automatic Toll Collection System

Algorithm

- Step 1: Entry IR Sensor detects vehicle; activate USB Camera.
- Step 2: Capture vehicle front image at 1280×720 resolution.
- Step 3: Pre-process frame using OpenCV (grayscale, Gaussian blur, Sobel edge detection, Otsu thresholding, morphological operations).
- Step 4: Localise license plate region via contour extraction and aspect ratio filtering (ratio 2.5–7.0).
- Step 5: Segment individual characters from the cropped plate region.
- Step 6: Apply Tesseract OCR (--psm 8, --oem 1) to recognise characters and extract registration number.
- Step 7: Display detected number plate and owner details on the web interface.
- Step 8: Query database for owner GPS coordinates; apply Haversine formula to compute distance to toll plaza.
- Step 9: If distance ≤ 10 km $\rightarrow$ No Toll; else $\rightarrow$ Deduct toll from linked account and record transaction.

- Step 10: Open gate via Servo Motor; update I2C 16x2 LCD Display and LED Indicator; sound Buzzer.
- Step 11: Exit IR Sensor detects vehicle departure; close gate and reset system.

Implementation

Image Processing with OpenCV

OpenCV handles all image processing stages prior to Tesseract OCR. Upon USB Camera image capture (1280×720, 30 fps), the frame passes through: Grayscale Conversion to reduce computational complexity; Gaussian Blur to suppress high-frequency noise; Sobel Edge Detection to highlight plate boundaries; Otsu Thresholding to convert to binary for contour analysis; Morphological Operations (dilation and erosion) to fill character gaps and remove noise; and Contour & Aspect Ratio Filtering to isolate the license plate region.

Character Recognition with Tesseract OCR

Tesseract OCR v4.1 is configured with `--psm 8` (single word mode) and `--oem 1` (LSTM-only engine), with an alphanumeric character whitelist to prevent punctuation misclassifications. Tesseract operates using a 2-pass recognition method: Pass 1 performs static classification using trained character image patterns; Pass 2 applies an adaptive model to improve accuracy on ambiguous characters.

Post-processing rules constrain Tesseract output to valid Indian plate patterns (two letters + two digits + two letters + four digits for standard plates). Common misclassifications on Indian plates — such as ‘B’ as ‘3’ and ‘D’ as ‘Q’ — are corrected by these rules.

Web Interface

A critical feature is that the detected number plate is displayed on a locally-hosted Python Flask web interface before the gate opens. The interface presents the cropped number plate image, extracted registration number, owner name and address from the database, and transaction status. Only after confirmation does the system issue the Servo Motor command to open the gate, ensuring operator visibility, transaction auditability, and prevention of unauthorised gate actuation.

Hardware Interfacing

All peripherals are interfaced with the Raspberry Pi 4B via GPIO pins. Entry and Exit IR Sensors use Python interrupt-based callbacks. The Servo Motor is driven via PWM output (0° = gate closed; 90° = gate open). The I2C 16x2 LCD Display communicates over the I2C bus (SDA/SCL pins). LED Indicator and Buzzer are driven via GPIO output pins. The complete Python control script coordinates all sensor readings, motor actuation, display updates, buzzer signals, and database queries concurrently.

RESULTS AND DISCUSSION

Table I. System Performance – Tesseract OCR with OpenCV Pre-processing

Metric	Precision	Recall	F1-Score
Tesseract OCR	90.81%	83.97%	87.26%

The system was evaluated on 120 real Indian vehicle number plates photographed in outdoor toll-like conditions, including daytime (40 images), night with artificial light (40 images), and angled/occluded plates (40 images).

The complete end-to-end pipeline operated correctly, with the full pipeline from Entry IR trigger to gate open completing in under 2 seconds per vehicle. The toll decision logic (local vs. non-local) was correctly applied in 100% of successful recognition cases.

Table II. Detailed Performance Under Varied Test Conditions

Test Condition	Plates Tested	Detection Acc.	OCR Acc.
Good Lighting	40	97.5%	93.2%
Low Lighting	40	87.5%	82.1%
Angled / Occluded	40	80.0%	74.6%
Overall	120	88.3%	83.3%

Table III. Gate Response Latency (30 Trials)

Metric	Min	Max	Avg	Std Dev
IR Trigger → Gate Open (s)	1.41	1.98	1.74	0.18
OCR Processing Time (s)	0.62	1.10	0.81	0.12
Toll Decision Accuracy	—	—	100%	—

Comparison with Existing Systems

Table IV. Comparison with Existing ANPR and Toll Automation Systems

System	Platform	OCR Acc.	Gate Ctrl	Toll Logic	Cost
Puranic et al. [10]	PC	80.8%	No	No	High
Shidore & Narote [12]	PC	79.84%	No	No	High
Palekar et al. [15]	RPi	~82%	No	No	Low
Proposed System	RPi 4B	90.81%	Yes	Yes	Low

As shown in Table IV, the proposed system surpasses all compared methods in OCR accuracy while being the only embedded solution to integrate both hardware gate control and automated toll decision logic on a low-cost platform.

Hardware Setup

The assembled hardware prototype integrates the Raspberry Pi 4B as the central controller, with Entry and Exit IR Sensors at the gate entry and exit points. The Logitech C270 USB Camera captures vehicle front images at 0.5–1 m distance. The Servo Motor actuates the gate barrier, and the I2C 16x2 LCD Display, LED Indicator, and Buzzer are connected via GPIO and I2C interfaces. Fig. 3 shows the complete hardware setup.

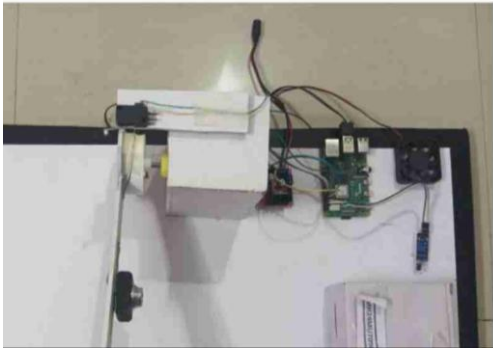


Fig. 3. Hardware Setup – Number Plate Detection and Automatic Toll Collection System

CONCLUSION AND FUTURE SCOPE

Conclusion

This paper presented a complete Number Plate Detection and Automatic Toll Collection System built on Raspberry Pi 4B, using OpenCV for image processing and Tesseract OCR (v4.1, --psm 8, --oem 1) for character recognition. The system employs a three-table database schema with Haversine-based distance computation for the 10 km toll-exemption rule. A web interface displays the detected number plate before gate actuation, adding operator visibility and auditability. Tesseract OCR achieved 90.81% precision and 83.97% recall, with end-to-end operation completing in under 2 seconds per vehicle. The proposed system outperforms prior Indian ANPR implementations [10,12] in accuracy and is the only embedded solution to integrate complete toll transaction logic with hardware gate control.

Limitations

- Camera quality significantly impacts OCR accuracy; low-quality cameras produce degraded plate images.
- Performance degrades under extreme illumination variations, heavy rain, or fog conditions.
- Current implementation processes one vehicle per gate at a time; multi-lane operation requires parallel processing.
- Partially occluded or damaged number plates may not be correctly recognised.
- The validation dataset (120 images) requires broader field trials for full generalisation.

Future Scope

- AI-powered plate recognition using YOLOv8 or TensorFlow Lite for higher accuracy on degraded images.
- Cloud integration for centralised toll data storage, real-time analytics, and remote monitoring.
- Integration with FASTag APIs and digital wallets for seamless nationwide payment.
- Multi-lane expansion with parallel Raspberry Pi nodes for full smart toll plaza deployment.
- SMS/Email notifications to vehicle owners for toll receipts and transaction alerts.
- Solar-powered operation for sustainable and remote-location deployment.
- Integration with government vehicle registration databases for enhanced owner verification.

ACKNOWLEDGEMENTS

The authors sincerely thank Dr. R. P. Patil, Guide, Department of E&TC, Sinhgad College of Engineering, for his invaluable guidance, patience, and continuous support. The authors also thank Dr. M. B. Mali, Head of Department, E&TC, Sinhgad College of Engineering, for his encouragement and inspiration. This work was made possible by the support of our families and friends.

REFERENCES

1. M. Fang, C. Liang, and X. Zhao, "A method based on rough set and SOFM neural network for car plate character recognition," in Proc. WCICA 2004, vol. 5, pp. 4037–4040, IEEE.
2. N. A. Jusoh, J. M. Zain, and T. A. A. Kadir, "Enhancing thinning method for Malaysian car plates recognition," in Proc. ICICIC'07, 2nd Int. Conf., pp. 378–378, IEEE, 2007.
3. E. R. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing," in Proc. ICIP-94, vol. 2, pp. 301–305, IEEE, 1994.
4. P. Hurtik and M. Vajgl, "Automatic license plate recognition in difficult conditions — Technical report," in Proc. IFSA-SCIS 2017, pp. 1–6.
5. S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 311–325, Feb. 2013.
6. P. Swaroop and N. Sharma, "An Overview of Various Template Matching Methodologies in Image Processing," Int. J. Comput. Appl., vol. 153, no. 10, Nov. 2016.
7. L. Kodwani and S. Meher, "Automatic License Plate Recognition in Real Time Videos using Visual Surveillance Techniques," ISSN 2320-8945, vol. 1, no. 6, 2013.
8. R. Islam, K. F. Sharif, and S. Biswas, "Automatic Vehicle Number Plate Recognition Using Structured Elements," in Proc. IEEE SPCCON, Dec. 2015, pp. 44–48.
9. M. T. Qadri and M. Asif, "Automatic Number Plate Recognition System for Vehicle Identification using Optical Character Recognition," in Proc. IEEE, 2009.
10. A. Puranic, D. K. T, and U. V, "Vehicle Number Plate Recognition System: A Literature Review and Implementation using Template Matching," Int. J. Comput. Appl., vol. 134, no. 1, Jan. 2016.
11. B.-G. Han, J. T. Lee, K.-T. Lim, and Y. Chung, "Real-Time License Plate Detection in High Resolution Videos Using Fastest Available Cascade Classifier and Core Patterns," ETRI J., vol. 37, no. 2, Apr. 2015.
12. M. M. Shidore and S. P. Narote, "Number Plate Recognition for Indian Vehicles," IJCSNS, vol. 11, no. 2, Feb. 2011.
13. A. O. Agbeyangi, O. A. Alashiri, and A. E. Otunuga, "Automatic Identification of Vehicle Plate Number Using Raspberry Pi," in Proc. IMCECS, Abeokuta, Nigeria, pp. 1–8, 2020.
14. A. Firasanti et al., "License Plate Detection Using OCR Method with Raspberry Pi," in Proc. IEEE Conf. Elect. Eng., Bekasi, Indonesia, pp. 1–6, 2021.
15. R. R. Palekar et al., "Real Time License Plate Detection Using OpenCV and Tesseract," in Proc. ICCSP, India, pp. 2111–2114, 2017.