International Journal of Research in Engineering, Science and Management
Volume 9, Issue 3, March 2026
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing | www.resaim.com

Automatic Number Plate Recognition Using Raspberry Pi

Faiz Muzaffar Ansari[1*], Kartik Anil Jakhere[1], Prathamesh Avinash Palande[1], Huzaifa Shakeel Ansari[1], Madhuri Pramod Rodge[1]

[1]*Department of Electronics and Telecommunication Engineering, Shivajirao S. Jondhle College of Engineering and Technology, Asangaon, India*

***Abstract*: Automatic Number Plate Recognition (ANPR) is an important technology used for automatically detecting and reading vehicle number plates using image processing techniques. It plays a major role in modern traffic management systems, parking automation, and security applications. In this project, a low-cost and efficient ANPR system is developed using Raspberry Pi 4 Model B+ along with server integration for data storage and remote access. The system captures vehicle images using a Raspberry Pi Camera Module and processes them using OpenCV. Image preprocessing techniques such as grayscale conversion, edge detection, and contour detection are used to identify the number plate region. After detection, Optical Character Recognition (OCR) is applied using Tesseract to extract the vehicle number. The extracted data is displayed on a TFT display (ST7789) and also sent to a server using HTTP requests. A server-based system is implemented using FastAPI, which stores detected number plates along with timestamps and images. This allows remote monitoring and data management through a web interface. The system works in real-time and provides good accuracy under proper lighting conditions. The main advantage of this system is its low cost, portability, and ability to integrate with cloud or local servers. It is suitable for applications such as parking systems, toll collection, and security surveillance, especially in small-scale environments.**

***Keywords*: ANPR, OpenCV, OCR, Raspberry Pi, Server Integration.**

1. Introduction

In today's world, the number of vehicles on the road is increasing very rapidly due to population growth and urban development. This increase creates many challenges in traffic management, parking control, and security monitoring. Managing such a large number of vehicles manually is very difficult and inefficient. Manual checking of vehicle number plates is time-consuming, error-prone, and not practical in busy areas such as toll booths, parking lots, highways, and city roads. It can also lead to delays, human mistakes, and security risks. Because of this, there is a strong need for automated systems that can handle these tasks efficiently and accurately.

Automatic Number Plate Recognition (ANPR) is a technology that helps in automatically detecting and reading vehicle registration numbers using cameras and image processing techniques. It reduces human effort and increases the speed and accuracy of vehicle identification. ANPR systems are widely used in applications such as traffic law enforcement, toll collection, parking management, border control, and surveillance systems. These systems help in tracking vehicles, identifying rule violations, and maintaining proper records without human intervention.

The basic working of an ANPR system includes capturing an image of a vehicle, detecting the number plate region, and extracting the text using Optical Character Recognition (OCR). This process involves several image processing steps such as grayscale conversion, noise removal, edge detection, and contour detection. The accuracy of the system depends on factors like lighting conditions, camera quality, angle of the vehicle, and clarity of the number plate.

Most advanced ANPR systems available in the market are expensive and require high-performance hardware and complex software, including deep learning models. While these systems provide high accuracy, they are not suitable for small-scale applications or for use in developing regions due to their high cost and complexity. Therefore, there is a need for a cost-effective and simple solution that can perform the same task with acceptable accuracy and reliability.

In this project, we have developed a low-cost ANPR system using Raspberry Pi, which is a compact and affordable embedded platform. The system uses a camera to capture images of vehicles and processes them using OpenCV to detect the number plate. Tesseract OCR is used to extract the text from the detected plate. The use of open-source tools makes the system flexible and easy to modify.

In addition to number plate recognition, a server-based system is also integrated using FastAPI, which allows storing and accessing the detected data remotely. The Raspberry Pi sends the detected number plate information along with images and timestamps to the server using HTTP requests. This server stores the data and allows it to be accessed through a web interface, making the system suitable for real-time monitoring and record management.

The main objective of this project is to design a system that is affordable, portable, and easy to implement while still providing reliable performance. The system is designed in such a way that it can be deployed in small-scale environments like parking areas, residential societies, and small toll systems. This

*Corresponding author: ansarifaiz812@gmail.com

project also helps in understanding the practical implementation of image processing, embedded systems, and server communication, which are important areas in modern engineering applications.

2. Literature Review

Many researchers have worked on Automatic Number Plate Recognition (ANPR) systems using different approaches and technologies. These systems mainly focus on improving accuracy, speed, and cost-effectiveness.

Agarwal and Pardakhe [1] developed an ANPR system using Raspberry Pi, which shows that low-cost hardware can be effectively used for real-time number plate detection. Their work highlights the importance of using embedded systems for portable and affordable solutions.

Kulkarni [2] studied different electronic toll collection systems and proposed a system for capturing and processing vehicle data. This work shows how ANPR can be integrated into real-world applications like toll collection to improve efficiency and reduce manual work.

Fajas et al. [3] presented a system specifically designed for Indian standard number plates. Their work focuses on handling variations in number plate formats, which is a major challenge in countries like India. This study is useful for understanding region-specific issues in ANPR systems.

Sarfaraz et al. [4] developed a license plate recognition system for Saudi Arabian vehicles. Their work demonstrates how ANPR systems can be adapted for different countries with different plate formats and designs.

Jain et al. [5] proposed an ANPR system using OpenCV, focusing on image processing techniques such as edge detection and contour analysis. Their approach shows that reliable results can be achieved using traditional image processing methods without requiring complex machine learning models.

Kumthekar et al. [6] implemented a Raspberry Pi-based number plate recognition system, which proves that embedded platforms can handle real-time processing tasks effectively. Their work also supports the idea of building low-cost and compact ANPR systems.

From the above studies, it is clear that ANPR systems can be implemented using both advanced and simple techniques. However, many existing systems either require high computational power or lack integration with modern data management systems. In this project, we aim to develop a low-cost ANPR system using Raspberry Pi and integrate it with a server using FastAPI for real-time data storage and remote access. This combination makes the system more practical and useful for real-world applications.

3. Methodology

The proposed Automatic Number Plate Recognition (ANPR) system follows a step-by-step process to detect and recognize vehicle number plates in real-time. The methodology is divided into different stages, starting from image capture to data storage on the server.

1) Image Capture

The system uses a Raspberry Pi Camera Module to capture real-time images of vehicles. The picamera2 library is used to continuously capture frames for processing.

2) Image Preprocessing

The captured image is converted into grayscale to reduce complexity. Noise is removed using filtering techniques, and edge detection is applied to highlight the boundaries of objects in the image.

3) Number Plate Detection

After preprocessing, contour detection is used to identify possible regions that may contain the number plate. The system filters these regions based on size, shape, and aspect ratio to accurately locate the number plate.

4) Character Recognition (OCR)

The detected number plate region is cropped and passed to the Tesseract OCR engine using the pytesseract library. This step extracts the alphanumeric characters from the plate image.

5) Display Output

The recognized number plate text is displayed in real-time on the TFT display (ST7789), providing immediate feedback.

6) Data Transmission

The extracted number plate text, along with the captured image and timestamp, is sent to the server using HTTP requests. The requests library is used for communication between the Raspberry Pi and the server.

7) Data Storage and Access

On the server side, FastAPI receives the data and stores it in a database or file system. The stored data can be accessed through a web interface for monitoring and record-keeping.

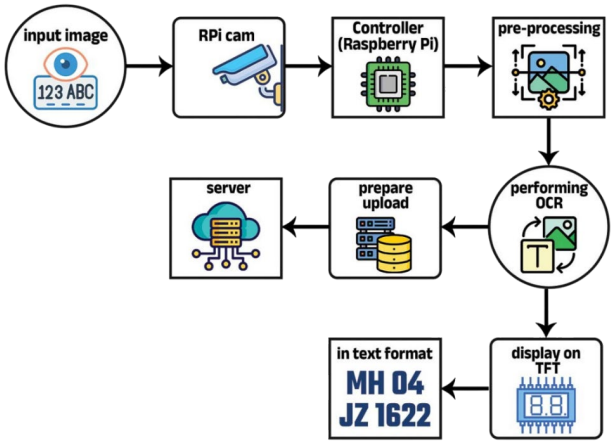

Fig. 1. Block diagram of ANPR

4. System Architecture and Design

The proposed Automatic Number Plate Recognition (ANPR) system is designed as a combination of an embedded system and a server-based system. The architecture is divided into two main parts: the client side (Raspberry Pi system) and the server side (FastAPI-based system). Both parts work together to provide real-time detection, display, and storage of vehicle number plate data.

A. Client Side (Raspberry Pi System)

The client side is built using a Raspberry Pi 4 Model B+, which acts as the main processing unit of the system. It is connected to a Raspberry Pi Camera Module and a TFT display (ST7789).

1) Image Capture Module

The Raspberry Pi Camera Module is used to capture real-time images of vehicles. The picamera2 library is used to control the camera and capture frames continuously. The captured images serve as input for further processing.

2) Image Processing Module

The captured image is processed using OpenCV. First, the image is converted into grayscale to simplify processing. Then, noise is reduced using filtering techniques, and edge detection is applied to highlight important features.

3) Number Plate Detection Module

After preprocessing, contour detection is used to identify possible number plate regions. The system filters contours based on size, shape, and aspect ratio to locate the number plate accurately.

4) OCR (Text Recognition) Module

Once the number plate is detected, the region is cropped and passed to Tesseract OCR using the pytesseract library. This module extracts the alphanumeric characters from the plate image.

5) Display Module

The extracted number plate text is displayed in real-time on the TFT display (ST7789). This provides immediate visual feedback to the user.

6) Data Transmission Module

After recognition, the system sends the extracted number plate text, along with the captured image and timestamp, to the server using HTTP requests. The requests library is used for communication.

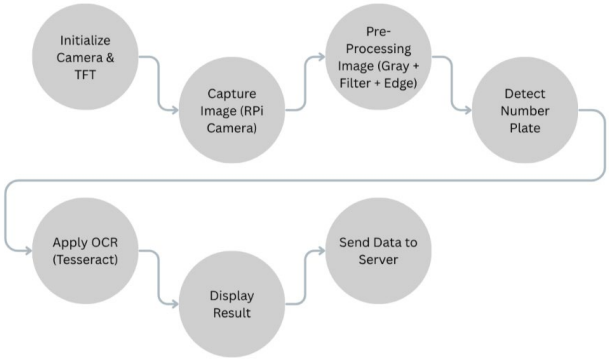

Fig. 2. Block diagram of the client-side

B. Server Side (FastAPI System)

The server side is developed using FastAPI, which provides a lightweight and fast backend for handling data.

1) API Handling Module

The FastAPI server receives incoming data from the Raspberry Pi through REST API endpoints. It handles requests such as uploading number plate data and retrieving stored records.

2) Data Storage Module

The received data, including number plate text, images, and timestamps, is stored in a local database or file system. This allows maintaining a history of detected vehicles.

3) Web Interface Module

The server provides a web-based interface where users can view the detected number plates and related data. This enables remote monitoring of the system from any device connected to the network.

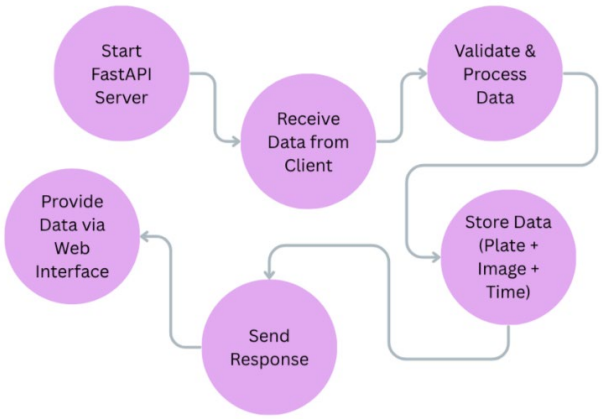

Fig. 3. Block diagram of the server-side

5. Hardware

The proposed ANPR system uses low-cost and easily available hardware components to achieve real-time number plate detection and recognition.

A. Raspberry Pi 4 Model B

Raspberry Pi 4 Model B+ is the main processing unit of the system. It acts as a mini computer that performs all operations such as image processing, number plate detection, OCR processing, and communication with the server. It is equipped with a powerful processor, sufficient RAM, and GPIO pins, which makes it suitable for real-time embedded applications. The Raspberry Pi runs the Python program that controls the entire ANPR system.

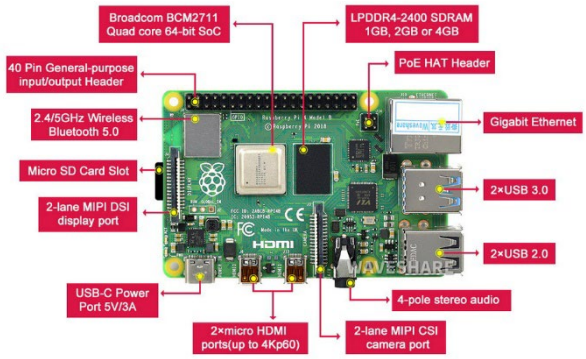

Fig. 4. Raspberry Pi 4 Model B

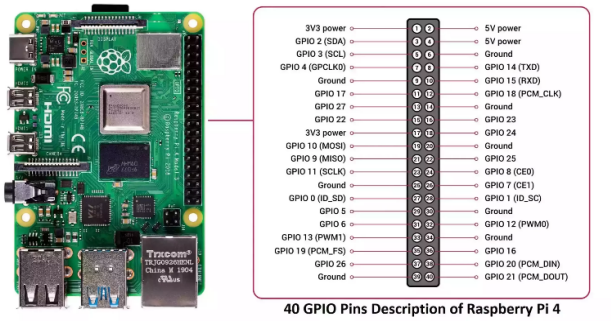

Fig. 5. Raspberry Pi 4 Model B Pinout

B. Raspberry Pi Camera Module Rev 1.3

The Raspberry Pi Camera Module is used to capture images of vehicles. It is connected to the Raspberry Pi through the CSI (Camera Serial Interface) port, which allows fast data transfer. The camera captures real-time frames that are used for number plate detection. Good image quality is important because it directly affects the accuracy of OCR and plate recognition.

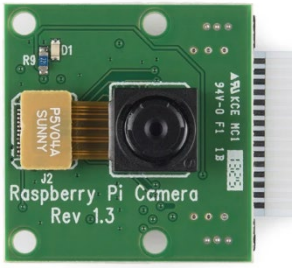

Fig. 6. Raspberry Pi Camera Module Rev 1.3

C. TFT Display (ST7789)

The ST7789 TFT display is used to show the output of the system. It displays the detected number plate text in real-time. The display is connected to the Raspberry Pi using the SPI communication interface. It provides a compact and efficient way to monitor the system without needing an external monitor.

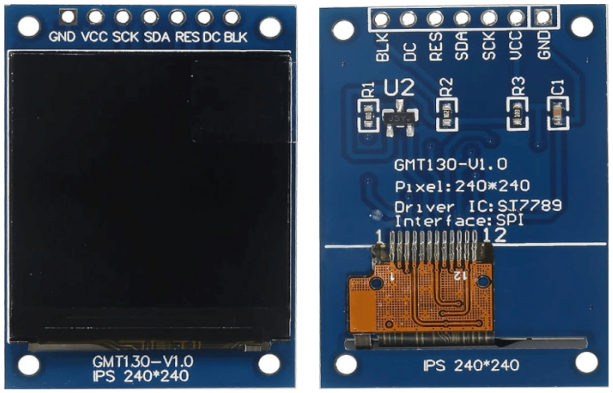

Fig. 7. TFT Display (ST7789)

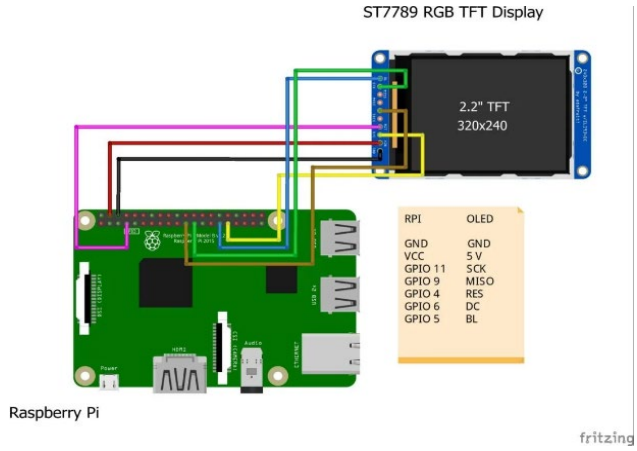

Fig. 8. RPi and ST7789 connection

6. Software

The system is developed using Python programming language along with various libraries that support image processing, OCR, and server communication.

A. Python Language & Thonny IDE

Python is used as the main programming language because it is simple, easy to understand, and has strong support for image processing libraries. Thonny IDE is used to write, edit, and run Python programs on the Raspberry Pi. It is beginner-friendly and suitable for embedded development.

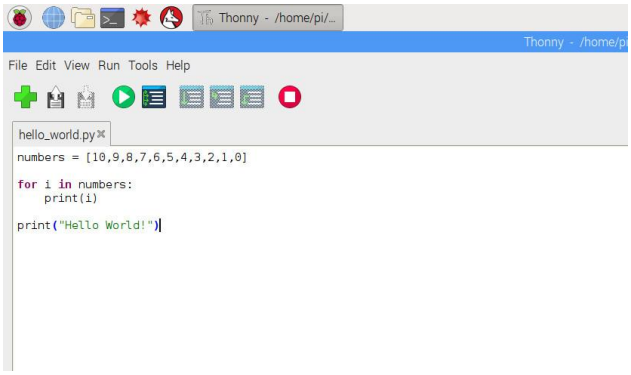

Fig. 9. Thonny IDE

B. OpenCV

OpenCV is the core library used for image processing. It is used for tasks such as converting images to grayscale, applying filters, detecting edges, and finding contours. These operations help in identifying the number plate region in the image.

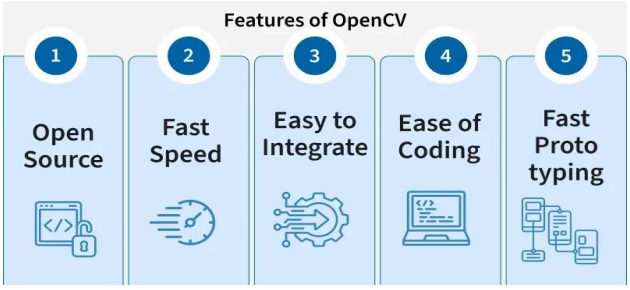

Fig. 10. OpenCV features

Fig. 11. OpenCV projects

C. *NumPy*

NumPy is used for handling image data in the form of arrays. It supports fast mathematical operations and is essential for processing images efficiently in OpenCV.

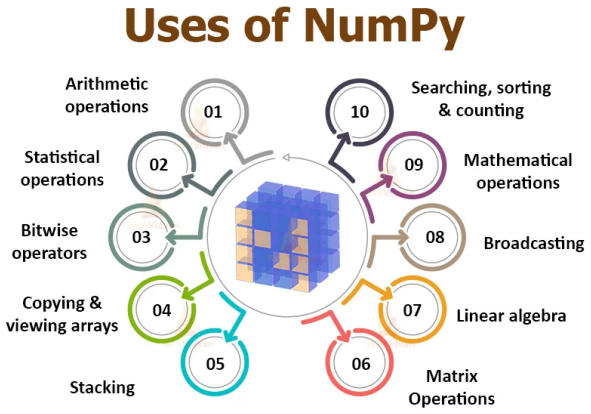

Fig. 12. Uses of NumPy

D. *Tesseract OCR (pytesseract)*

Tesseract OCR is used to extract text from the detected number plate image. The pytesseract library acts as a bridge between Python and the Tesseract engine. It converts the image of the number plate into readable text.

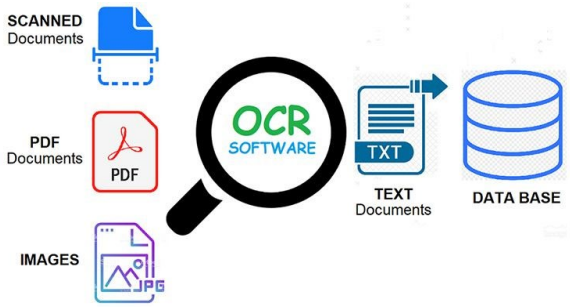

Fig. 13. OCR

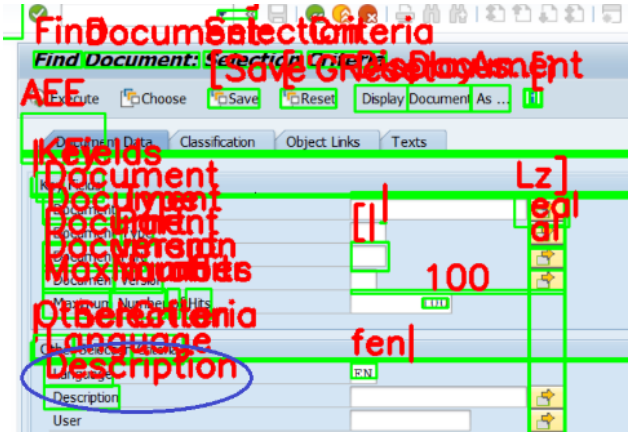

Fig. 14. Example of OCR

E. *PiCamera2*

The picamera2 library is used to control the Raspberry Pi camera. It helps in capturing images and video frames efficiently for real-time processing.

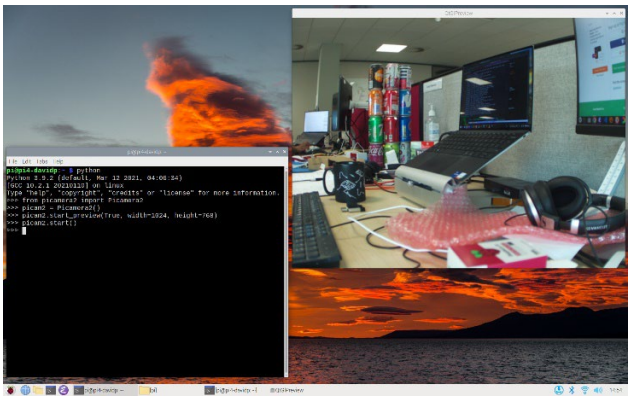

Fig. 15. picamera2 use

F. *FastAPI*

A lightweight and high-performance web framework used to build the server. It handles incoming data from the client and provides APIs for storing and accessing records.

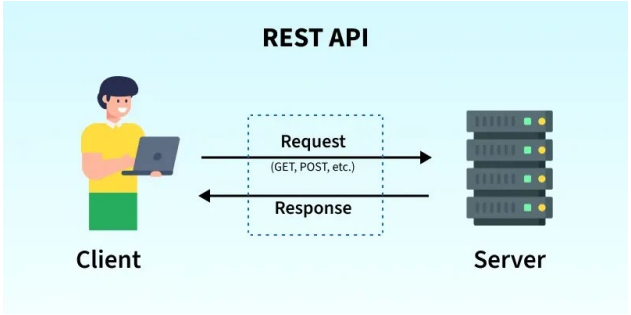

Fig. 16. FastAPI

G. *Additional Libraries and Tools*

i. Pillow (PIL)
ii. Requests
iii. Imutils

7. Results and Discussions

The proposed ANPR system was successfully implemented using Raspberry Pi and tested under different conditions. The system was able to capture vehicle images, detect number plates, extract text using OCR, and send the data to the server in real-time.

During testing, the system performed well in normal lighting conditions. The number plate detection using OpenCV worked effectively when the image was clear and properly aligned. The Tesseract OCR was able to recognize most of the characters correctly, and the detected number was displayed on the TFT screen without delay.

The server system also worked properly by receiving the data from the Raspberry Pi and storing it along with images and timestamps. The stored data could be accessed through the web interface, which confirmed successful communication between the client and server.

However, some limitations were observed. The system performance decreases in low light conditions or when the image is blurred. Detection may also fail if the number plate is tilted or partially hidden. OCR errors can occur when characters are not clearly visible.

Overall, the system provides satisfactory performance for a low-cost setup and is suitable for small-scale applications such as parking management and entry monitoring.

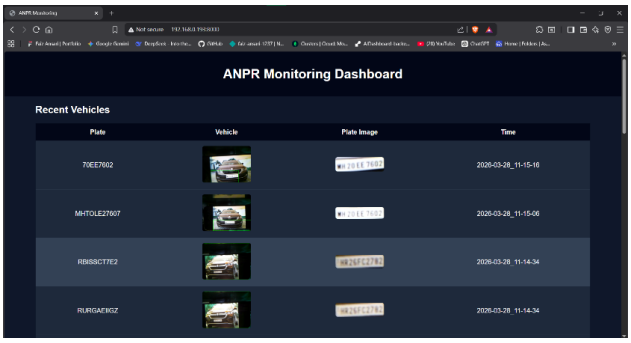

Fig. 17. Server

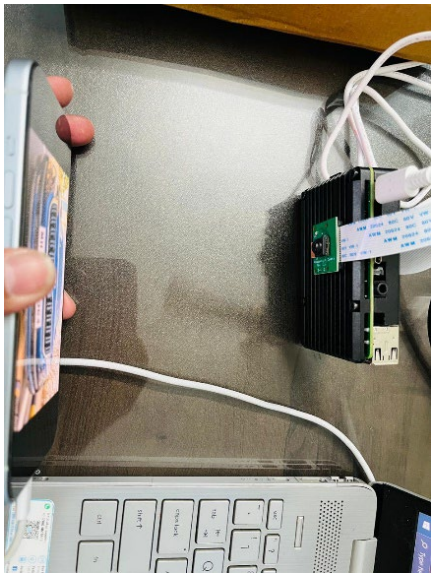

Fig. 18. Frame capture

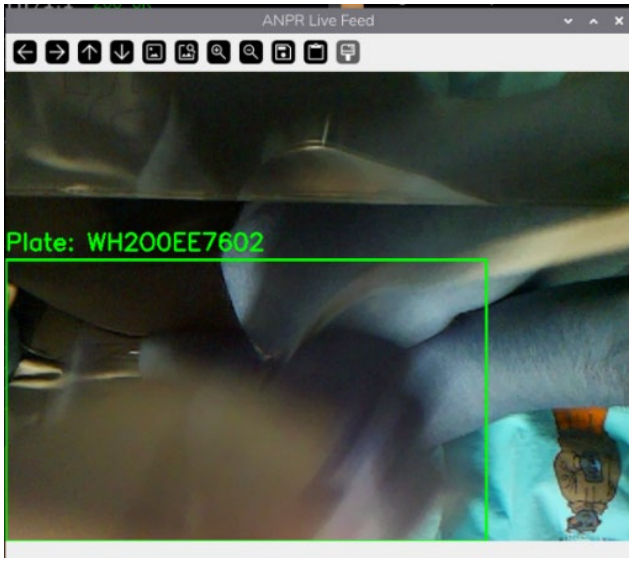

Fig. 19. Live feed

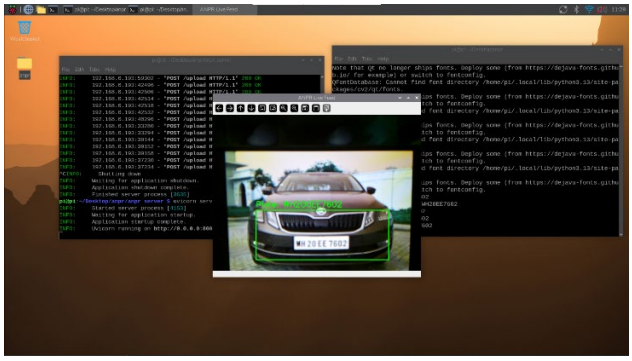

Fig. 20. ANPR system control

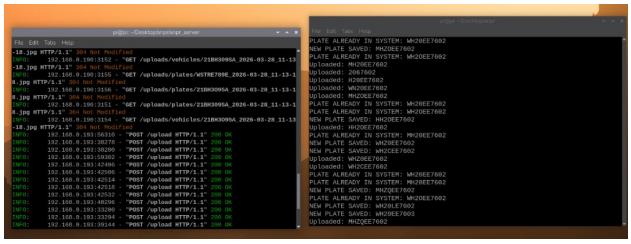

Fig. 21. Backend

A. Applications

The proposed ANPR system can be used in the following applications:

i. Parking management systems for automatic vehicle entry and exit tracking
ii. Toll collection systems for automated billing and reduced manual work
iii. Traffic monitoring systems to identify vehicles and manage traffic flow
iv. Law enforcement systems to detect rule violations and unauthorized vehicles
v. Security and surveillance systems for maintaining vehicle records
vi. Residential societies and restricted areas for access control

B. Advantages

The proposed system offers several advantages that make it suitable for practical use. First, it is a low-cost solution because it uses Raspberry Pi and open-source software, which makes it affordable compared to commercial ANPR systems. The system is also compact and portable, allowing it to be easily installed in different locations without requiring complex infrastructure.

Another major advantage is real-time processing. The system can capture images, detect number plates, and display results instantly on the TFT screen. Additionally, the integration of a FastAPI-based server allows the system to store data and provide remote access through a web interface. This makes it more useful for monitoring and record-keeping purposes.

The use of Python and simple libraries such as OpenCV and Tesseract makes the system easy to develop, understand, and modify. This flexibility allows future improvements and customization based on requirements.

C. Limitations

Despite its advantages, the system has some limitations that affect its performance in certain conditions. One of the main limitations is dependency on lighting conditions. The system performs well in good lighting but shows reduced accuracy in low light or night conditions due to poor image quality.

Another limitation is sensitivity to image quality and positioning. If the number plate is blurred, tilted, partially hidden, or dirty, the system may fail to detect it correctly. OCR accuracy is also affected by variations in font style, size, and spacing of characters on the number plate.

Since the system is based on basic image processing techniques, it may not perform as accurately as advanced systems that use deep learning models. It may face challenges in complex environments such as crowded scenes or multiple vehicles in a single frame.

8. Conclusion

In this paper, a low-cost Automatic Number Plate Recognition (ANPR) system using Raspberry Pi has been successfully designed and implemented. The system is capable of capturing vehicle images, detecting number plates using OpenCV, and extracting text using Tesseract OCR. The detected number plate is displayed on a TFT screen and also sent to a FastAPI-based server for storage and remote access. This integration of embedded system and server makes the system more efficient and suitable for real-time applications.

The system performs well under normal lighting conditions and provides satisfactory accuracy for small-scale applications such as parking management and entry monitoring. It is compact, portable, and easy to implement, making it a practical solution for low-cost environments.

However, there is scope for further improvement in the system. The performance can be enhanced by using higher quality cameras and better lighting conditions. Advanced techniques such as deep learning-based object detection and recognition can be applied to improve accuracy in complex scenarios. Additionally, cloud integration and mobile application support can be added for better accessibility and scalability of the system.

Acknowledgment

The authors acknowledge the guidance and the institutional support of Dr. (Mrs.) Madhuri P. Rodge, Head of Department, and Dr. (Mrs.) Geetha K. Jayaraj, Principal, Shivajirao S. Jondhle College of Engineering and Technology, Asangaon, University of Mumbai (B.E. programme, 2025-26).

References

[1] A. Agrawal and N. Pardakhe, “Automatic license plate recognition using Raspberry Pi,” in *Proc. International Interdisciplinary Conference on Science, Technology, Engineering, Management, Pharmacy and Humanities*, Singapore, Apr. 22–23, 2017, p. 4.

[2] A. H. Kulkarni, “Study of different electronic toll collection systems and proposed toll snapping and processing system,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 3, pp. 1357–1360, Mar. 2014.

[3] F. Fajas *et al.*, “Automatic number plate recognition for Indian standard number plates,” in *Proc. IEEE International Congress on Ultra-Modern Telecommunications and Control Systems*, 2012, pp. 1026–1028.

[4] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, “Saudi Arabian license plate recognition system,” in *Proc. International Conference on Geometric Modeling and Graphics*, 2003, pp. 36–41.

[5] P. Jain, N. Chopra, and V. Gupta, “Automatic license plate recognition using OpenCV,” *International Journal of Computer Applications Technology and Research*, vol. 3, no. 12, pp. 756–761, 2014.

[6] A. V. Kumthekar, S. Owhal, S. Supekar, and B. Tupe, “Recognition of vehicle number plate using Raspberry Pi,” *International Research Journal of Engineering and Technology*, vol. 5, no. 4, 2018.