MINI PROJECT REPORT

on

Automatic License Plate Recognition using Raspberry Pi for Gate Control and Vehicle Monitoring

Submitted by:

Shwetha Iyer (1032211195)

Saanvi Kulkarni (1032210698)

Pranay Gupta (1032211057)

Under the Guidance of

Dr. Vaidehi Deshmukh

School of Electronics & Communication Engineering

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY, PUNE.
[2023-2024]

DEPARTMENT OF
ELECTRICAL AND ELECTRONICS ENGINEERING

Academic Year 2023-2024

CERTIFICATE

This is to certify that Shwetha Iyer, PRN No. 1032211195; Saanvi Kulkarni, PRN No. 1032210698; Pranay Gupta, PRN No. 1032211057 have successfully completed their Mini Project entitled "**Automatic License Plate Recognition using Raspberry Pi for Gate Control and Vehicle Monitoring**" and submitted the same during the academic year 2023-24.

(MiniProject batch Coordinator)

Dr. Parul Jadhav
(Program Director, DoEEE)

Date: 30.04.2023

Place: Pune

Table of Contents

Abstract		**4**
List of Figures		**5**
Abbreviations		**5**
1.	**Introduction**	**6**
1.1	Motivation	6
2.	**Review of Literature**	**6**
2.1	Literature Review	6
2.2	Aim and Objectives	7
3.	**System Development**	**8**
3.1	System Block Diagram	8
3.2	System Specifications	8
3.3	Challenges Involved	9
4.	**System Implementation**	**9**
4.1	System Design and Description	9
4.2	Flow Chart	11
5.	**Results and Analysis**	**12**
5.1	Results and Analysis of Implementation	12
6.	**Conclusion**	13
	References	**13**

ABSTRACT

A license plate number is a unique identifier of a vehicle and its owner. Automatic License Plate Recognition (ANPR) systems help to capture a plate number, extract the numbers on the plate and check the details of the owner. The increasing number of vehicles presents a challenge for security agents to manually identify and monitor them effectively. This paper describes the development of an ANPR system for gate control and vehicle monitoring. The system uses Raspberry Pi, OpenCV and camera to capture images of vehicles, detect license plates, and extract the alphanumeric characters. Built for real-time operation, it utilizes a trained YOLOv7 model for accurate license plate detection and EasyOCR for character recognition. The paper discusses hardware and software specifications, system design and implementation, challenges faced and experimental results. The results from testing show that the system performed well in most conditions. This cost-effective ANPR system holds promise for various applications such as gate control and access management, traffic monitoring, and aiding law enforcement in identifying stolen vehicles.

LIST OF FIGURES

Figure 3.1	Block Diagram of ALPR System using Raspberry Pi	8
Figure 4.2	Flowchart of ALPR Pipeline	11
Figure 5.1	(a) Results of Post-Detection Cropping (b) Tabulated List of Plates after OCR	13

ABBREVIATIONS

ALPR	Automatic License Plate Recognition
ANPR	Automatic Number Plate Recognition
GPIO	General Purpose Input/Output
OCR	Optical Character Recognition
YOLO	You Only Look Once
COCO	Common Objects in Context
API	Application Programming Interface
mAP	Mean Average Precision

1. INTRODUCTION

1.1 Motivation

With the rise in the number of vehicles, the task of efficiently identifying and monitoring them has become increasingly challenging for security agents. Automatic real-time identification of vehicles has become necessary due to its diverse applications, including traffic control, surveillance, parking and toll management, access control, ticket issuance, theft prevention and document verification. It has been widely considered as an important area of research with respect to modern automation and intelligent transportation systems. Such technology can be used for identifying vehicles that break traffic rules, monitoring the time of entry and exit in certain premises, registering information on parked vehicles, and facilitating the resolution of vehicle-related crimes.

This project is proposed to automate the detection and recognition of license plates of vehicles and expedite the tedious and time-consuming process of manually comparing number-plates with a database of vehicles of interest.

2. REVIEW OF LITERATURE

2.1 Literature Review

Many studies have addressed automated license plate recognition, alongside its solutions and challenges. The early approaches to ALPR focused on traditional computer vision techniques such as edge detection and contouring for both license plate detection and recognition. Although these solutions were simple, quick, and lightweight, they failed in complex scenarios and were often sensitive to noise, illumination and license plate variations.

S. Fakhar A. G., M. Saad H., A. Fauzan K., R. Affendi H., M. Aidil A (IJECE 2018) have presented a cost-effective ANPR system using Raspberry Pi and OpenCV. Image processing tasks such as desaturation, filtering, and character recognition are performed on the device. Experimental results show a 85% success rate with a 3-second processing delay.

Naveena Budda, K. Meenakshi, Padmavathi Kora, et al. (IJITEE 1029) have proposed an ANPR system using Python and OpenCV library. The system captures an image,

converts it to grayscale, removes noise, detects edges, finds contours, and displays the number plate area.

Newer solutions for ALPR have considered machine learning and deep-learning-based approaches. These solutions achieve high performance but consume more resources and processing power when compared to classical methods. In deep learning, the problem of ALPR can be considered to be a general object detection and a character recognition problem.

Lubna, Naveed Mufti and Syed Afaq Ali Shah's Automatic Number Plate Recognition: A Detailed Survey of Relevant Algorithms (Sensors 2021) provides a review of real-time ANPR algorithms and explores challenges like plate conditions, camera quality, and environmental factors, highlighting ANPR's role in intelligent transportation systems (ITS). The integration of ANPR with technologies like RFID and GPS is discussed, along with the influence of deep learning in computer vision.

Heshan Padmasiri, Jithmi Shashirangana, Dulani Meedeniya et al. (Sensors 2022) have introduced a novel approach for ALPR for resource-constrained edge computing environments. It addresses challenges such as nighttime operation without additional lighting, low-cost deployment, low-latency and energy efficiency for smart city applications.

2.2 Aim and Objectives

The aim of this project is to develop an automated system for license plate recognition to monitor vehicles using a camera and Raspberry Pi microprocessor. With the help of this number plate recognition system, one can recognize a license plate number, compare it with a list in a relevant database, and check the details of the vehicle owner. If the number matches with any of the number plates in the list, the barrier will open and give way to the vehicle. This project would aid in permitting and monitoring vehicle entry/exit at entrance gates or parking areas that have cameras.

The objective of this effort is to provide an efficient and convenient way to monitor vehicles. This includes: authorizing entrance for selected vehicles, collecting data related to vehicles (such as owner's name, entry/exit time), and identifying lost or stolen vehicles. The automatic number plate recognition system would function to capture an

image of the car, process the image to detect a number plate, compare the acquired plate number with a database of vehicles of interest, allow security agents to automatically verify the plate number of vehicles entering and exiting, and confirm the driver's identity and the vehicle's particulars through system-stored information. If there is a match found in the list of number plates, the barrier gate would open to admit the vehicle.

3. SYSTEM DEVELOPMENT

3.1 System Block Diagram

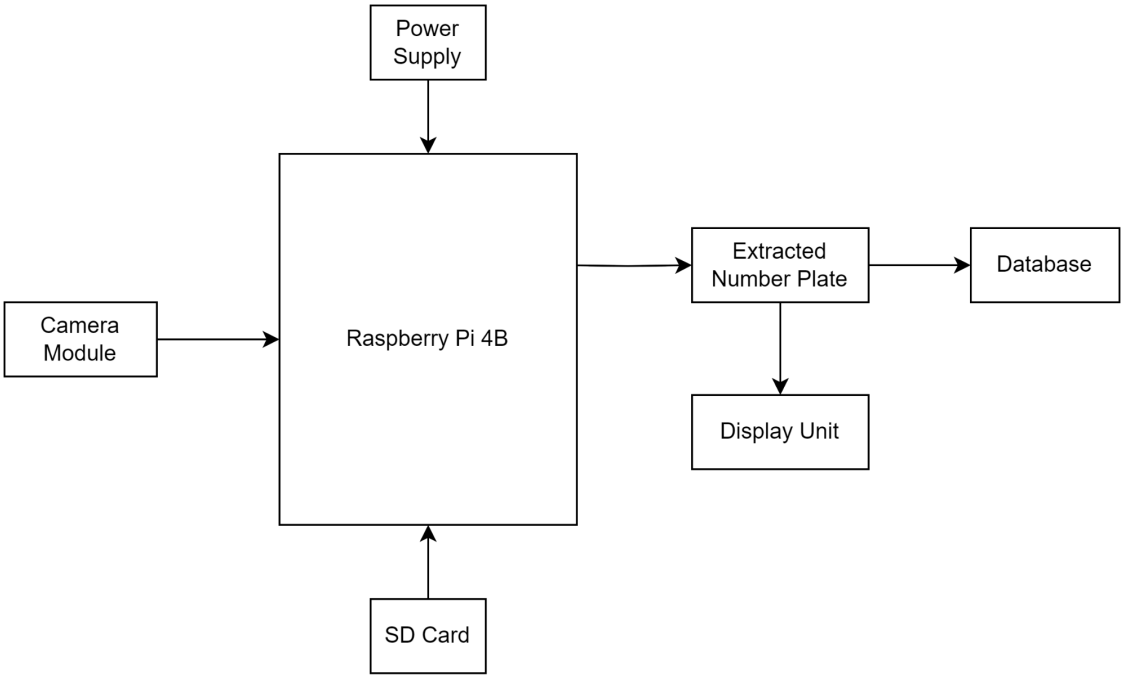

Figure 3.1 Block Diagram of ALPR System using Raspberry Pi

3.2 System Specifications

Hardware Specifications:

- Camera — Raspberry Pi 5MP Camera Module V2 (2592 x 1944 Image Size; 1080p30 and 720p60)
- Edge device — Raspberry Pi 4 Model B (4GB RAM; Quad-core Cortex-A72 processor; 64-bit SoC @ 1.5GHz; 40 pin GPIO header)
- Actuator — TowerPro SG90 Mini Servo Motor (3.0V~7.2V Operating Voltage, 180^{o} Rotation)

- Power Supply — 5V 3A USB-C

Software Specifications:

- Number Plate Detection — OpenCV, Fine-tuned YOLO v7 Model
- OCR and Digit Extraction — EasyOCR

3.3 Challenges Involved

1. Poor lighting, blurring, occlusions, angle variations and environmental conditions are factors that affect image quality and hence impact the performance of ALPR.
2. License plates may vary across regions or states, and they may even differ in size, font, colour, design, etc. The images used may be biased towards certain locations or plate formats, so training a model to accurately detect all license plate variations and localize it accurately can be challenging.
3. Managing a real-time database of vehicle information such as owner details and entry/exit times requires efficient and reliable storage, retrieval and updating mechanisms.
4. The OCR accuracy may struggle with skewed or damaged license plates, ambiguous characters or unclear fonts, low-resolution images and presence of noise or artifacts. For example it may misinterpret ‘0’ as ‘O’ and ‘1’ as ‘I’, or vice versa.
5. Applications like traffic monitoring and access control depend on the real-time performance of the ALPR system, which might be limited due to processing power and latency.
6. There are legal and ethical concerns regarding collection and storage of license plate data and owner information. It is important to implement data protection measures and comply with privacy regulations.

4. SYSTEM IMPLEMENTATION

4.1 System Design and Description

This ALPR project is designed to detect and recognize license plates of vehicles in real-time using a combination of hardware and software components. Here are the steps involved in system design:

1. *Data Acquisition and Annotation*

The project began with the creation of a custom dataset containing approximately 650 images of vehicles with Indian license plates. These images include 450 self-captured images and 200 images gathered from various online sources. This data collection effort also focused on capturing images to cover different variations such as diverse backgrounds (streets and parking lots), vehicle types (cars, trucks, two-wheelers, and buses), angles and lighting conditions.

Roboflow is a fast and easy-to-use computer vision tool used to organize, label and export image datasets. With the help of Roboflow, each image was manually annotated by drawing bounding boxes around the license plate area. These annotations were only for a single class called 'numberplate'. Automatic data augmentation techniques were applied via Roboflow to allow for model robustness and generalization. This expanded the dataset to over 1600 images through transformations like flipping, brightness adjustment and zooming. The final dataset was divided into training, validation and testing sets.

2. *Model Training and Refinements*

YOLOv7 is a state-of-the-art real-time model for object detection in images and videos, known for its speed and accuracy. It outperforms previous versions like YOLOv5 and YOLOv4, achieving an average precision of 56.8%. It has several advancements for improved feature integration, computation and training. This makes it useful and versatile for many real-time applications like image classification, face recognition, and autonomous driving. YOLOv7 also requires less computational power to run compared to other models, and it can easily be trained on small datasets.

Hence, the YOLOv7 model was chosen for license plate detection due to its accuracy and easy implementation. A workspace API on Roboflow was utilized to access the prepared dataset and train the model for 40 epochs. The trained model achieved a high level of accuracy, with precision (P) and recall (R) both at 0.991, and mean average precision (mAP) exceeding 0.9 for various confidence thresholds.

To address instances of multiple license plate detections in some images, adjustments were made to the YOLOv7 code to consider the bounding box with the highest confidence score for further cropping.

3. *Optical Character Recognition (OCR)*

   After detection, the images underwent preprocessing such as grayscale conversion and skew correction. Then OCR was employed to extract text from the cropped license plate images. This was done using EasyOCR, an open-source OCR library. The extracted text was refined further to eliminate unnecessary elements like symbols or extraneous text snippets.

4. *Hardware Setup and Deployment*

   Raspberry Pi 4 was chosen as the edge device due to its compact size and computational abilities. The hardware setup involved installing the OS, interfacing the camera module for real-time image capture, and installing necessary libraries/modules to execute the ALPR model.

4.2 Flowchart

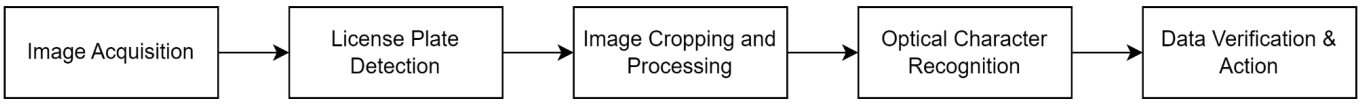

Figure 4.2 Flowchart of ALPR Pipeline

The process of license plate recognition can be executed in four stages:

1. **Image Acquisition:** The number plate image is captured with the help of a 5MP Raspberry Pi camera and stored in an SD card for further processing and recognition. OpenCV is utilized to directly access video frames captured by the camera. Once the image is captured, it undergoes processing for detection of the number plate.
2. **License Plate Detection:** The next step involves extracting the license plate from the captured image. For real-time object detection, one can use the YOLOv7 model, which is pre-trained on the COCO dataset. The model can be fine-tuned further on a custom dataset of license plates. It outputs a cropped region with a corresponding bounding box and confidence score. This image undergoes specific preprocessing like grayscale conversion, noise removal, segmentation, etc. to prepare it for OCR.
3. **Character Recognition:** The next step involves recognizing the corresponding characters/numbers on the license plate using a technology called Optical Character Recognition (OCR). This can be achieved using EasyOCR or PyTesseract modules in Python.

4. **Verification and Action:** After the text is detected and read from the number plate, it needs to be verified against some database, for example a CSV file. Additionally, automatic control can be implemented by connecting GPIO pins of Raspberry Pi to a motor controlling the barrier gate, or any load like LED, buzzer, etc.

5. RESULTS

5.1 Results and Analysis of Implementation

This study successfully implemented and demonstrated an ANPR system using a Raspberry Pi. The system captures images, runs a trained YOLOv7 model for object detection, identifies and crops license plates, applies OCR to extract the plate numbers and finally tabulates a list of license plates. The system performed best when capturing daytime images of plates within 2 meters of a forward-facing camera. The system's performance was considered based on accuracy, speed and power efficiency. Overall, the proposed system has shown notable robustness to variations in angle and illumination changes and achieved competitive results compared to traditional edge detection-based solutions.

Figure 5.1 (a) Results of Post-Detection Cropping

	A	B
1	**Filename**	**Number Plate**
2	numberplate0.jpg	MHI2DE142J
3	numberplate1.jpg	MH04JM8765
4	numberplate2.jpg	CHOIANOOI
5	numberplate3.jpg	KA03AB3289
6	numberplate4.jpg	HHOLAM1571
7		

Figure 5.1 (b) Tabulated List of Plates after OCR

6. CONCLUSION

This project successfully built an automatic number-plate recognition system using a Raspberry Pi for real-time license plate detection and character recognition. The implemented ANPR system effectively identifies and recognizes vehicle number plates, showing considerable potential for the automation of entry processes for enhanced security. However, ongoing research in ANPR presents a need for more robust algorithms across diverse plate formats and real-time testing scenarios. Future efforts into this project will target reducing processing time, enhancing accuracy for diverse license plates, and exploring alternative recognition technologies for improved accuracy and reliability.

REFERENCES

1. G, S. F. A., H, M. S., K, A., H, R., & A, M. A. G. (2019, June 1). *Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi*. International Journal of Electrical and Computer Engineering. https://doi.org/10.11591/ijece.v9i3.pp1805-1813
2. Lubna, L., Mufti, N., & Shah, S. A. A. (2021, April 26). *Automatic Number Plate Recognition:A Detailed Survey of Relevant Algorithms*. Sensors. https://doi.org/10.3390/s21093028
3. Padmasiri, H., Shashirangana, J., Meedeniya, D., Rana, O., & Perera, C. (2022, February 13). *Automated License Plate Recognition for Resource-Constrained Environments*. Sensors. https://doi.org/10.3390/s22041434
4. *B7919129219 - International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. (2022, April 27). International Journal of Innovative Technology and Exploring Engineering (IJITEE). http://dx.doi.org/10.35940/ijitee.B7919.129219
5. *Automatic Identification of Vehicle Plate Number using Raspberry Pi*. (2020, March 1). IEEE Conference Publication | IEEE Xplore. https://doi.org/10.1109/ICMCECS47690.2020.246983
6. Yeryomenko, O. (2022, December 27). *Automatic Number Plate Recognition with Raspberry Pi*. Medium.

https://medium.com/@alexey.yeryomenko/automatic-number-plate-recognition-with-raspberry-pi-e1ac8a804c79