Vehicle Authorization System Using Raspberry Pi

Mr. Suyash Gadekar[1], Ms. Aishwarya Patil[1], Mr. Aryan Pise[1], Assistant Prof. Mrs.A.A. Kasangottuwar[2]
[1]*UG Students, Department of Electronics and Telecommunication Engineering*
[2]*Assistant Professor, Department of Electronics and Telecommunication Engineering*
Department of Electronics and Telecommunication Engineering, P.E.S Modern College of Engineering, Shivajinagar, Pune-5

***Abstract* — This project introduces an affordable solution for automatic vehicle detection and access control using Raspberry Pi. It combines a camera module, OpenCV, and OCR to process live vehicle images and identify number plates. The system automates gate control and sends alerts, enhancing speed, accuracy, and scalability across multiple use cases.**

The growing demand for smart transportation and enhanced security in urban infrastructure has led to the development of intelligent vehicle access control systems. This project, titled "Automated Vehicle Access Control with Number Plate Recognition Using Raspberry Pi", presents a cost-effective and efficient solution for real-time vehicle identification and access management using license plate recognition.

A Raspberry Pi serves as the main controller and is connected to a camera to capture real-time images of vehicles. These images undergo preprocessing with OpenCV to improve quality and isolate the license plate area. OCR is then used to read and extract the characters from the plate. The recognized license plate number is matched against a preloaded database of authorized vehicles to determine access eligibility. Upon successful identification, the system triggers automated access actions such as opening a gate or logging entry details, and can also send SMS notifications to stakeholders using an API like Twilio. The entire process operates without human intervention, thereby improving accuracy, reducing response time, and enhancing the overall security and convenience of vehicle access management.

This project is scalable and applicable in a wide range of domains including gated communities, corporate campuses, toll booths, parking facilities, and law enforcement checkpoints. By integrating affordable hardware and open-source software tools, it offers a sustainable approach to modernizing transportation and security systems.

***Index Terms*- Raspberry Pi, Vehicle Access Control, License Plate Detection, Optical Character Recognition (OCR), Automated Gate System, OpenCV, Smart Transportation, Real-Time Image Processing, Intelligent Surveillance, IoT-Based Security System, Python Programming, Smart Parking System, Vehicle Authorization**

I. INTRODUCTION

The Smart Vehicle Authorization and Number Plate Detection the system is created to offer a smart, streamlined, and automated method for identifying vehicles based on their number plates. The system utilizes Optical Character Recognition (OCR) in conjunction with advanced image processing algorithms to extract and interpret alphanumeric data from vehicle number plates. It is built using the Raspberry Pi, a compact and cost- effective computing platform ideal for real-time embedded applications.

This project leverages the processing capabilities of Raspberry Pi to build a live system that can identify number plates instantly as vehicles approach. The system works by capturing images of approaching vehicles using a camera that's attached to the Raspberry Pi captures vehicle images for processing. The captured images are cleaned up and processed to improve visibility and focus specifically on the number plate.

After spotting the number plate, OCR methods are used to extract the text information from it. This extracted data is then cross-referenced with a local or cloud-based database containing registered vehicle and owner details. Upon verification, the system determines whether the vehicle is authorized to enter or exit a specified area (e.g., residential complex, office parking, toll booth).

Figure1.1: License Plate Recognition Using OpenCV with Raspberry Pi 5

- Real-time vehicle number plate recognition using Raspberry Pi and Python.
- Image enhancement techniques for accurate plate detection.
- Integration of OCR for alphanumeric extraction.
- Access control via automated verification against a registered database.
- Entry/Exit alerts through SMS or cloud notification.
- Scalable, low-cost deployment suitable for smart city and smart parking applications.
- Applications in traffic monitoring, parking management, gated communities, and law enforcement.

II. MOTIVATION

Vehicle theft has become a widespread concern in recent years. To combat this, our system is built to smartly identify and recognize vehicle number plates, identify authorized vehicles, and notify the owner with real-time alerts regarding vehicle arrival and departure. The goal is to create a smart solution that can be easily deployed in applications such as smart cities, parking management, gated communities, and law enforcement—offering improved security, convenience, and control.

This project ultimately aims to demonstrate how accessible, modern technologies can be integrated to solve real-world problems in urban mobility and security.

III. LITERATURE SURVEY SUMMARY

The paper discusses research projects focusing on Automatic Number Plate Recognition (ANPR) and IoT-based sensors in urban vehicle parking systems. It explores the development of an efficient ALPR system using Optical Character Recognition and Fuzzy Searching techniques, and proposes an Automatic License Plate Detection and Recognition (ALPDR) system. The paper also discusses the importance of biometric authentication in vehicle ignition and discusses the design of real-time applications for traffic monitoring and car locking systems. It also discusses digital vehicle identification for forensic, technical, and administrative purposes, and the need for an automatic vehicle registration system.

IV. GOAL OF WORK

The project aims to create an automated system using image processing and OCR technology to detect and recognize vehicle number plates, utilizing Raspberry Pi for cost-effective, real-time applications, enhancing security, reducing human intervention, and supporting smart cities and traffic surveillance.

V. PROBLEM STSTEMENT

In today's rapidly evolving urban environments, manual vehicle identification and monitoring methods are often inefficient, prone to human error, and lack real-time responsiveness. There is a critical need for a cost- effective, automated system that can accurately identify and authorize vehicles in areas such as high-security premises. Existing solutions may be expensive, complex to deploy, or lack integration with modern smart infrastructure. This project tackles the issue by building a Raspberry Pi-based vehicle access system that uses OCR and basic image processing to detect license plates automatically. It provides a flexible, low-power solution ideal for modern smart city needs.

VI. OBJECTIVES

- To build a smart system that can automatically detect and identify vehicle license plates using image processing methods and OCR technology.
- To implement the system using Raspberry Pi, making it compact, cost-effective, and suitable for real-time applications.
- To enhance security by verifying vehicles against a database and sending notifications to the owner upon vehicle entry or exit.
- To reduce human intervention and errors in vehicle access control and monitoring systems.
- To support applications in smart cities, parking lots, gated communities, and traffic surveillance with a scalable and efficient solution.

VII. MATERIAL TO BE USED

Hardware Requirements:

- Raspberry pi 4 Model B (2GB RAM) – Central processing unit
- Pi Camera Module – Captures images of vehicle number plates
- Power Supply (5V, 3A) – give powers to the Raspberry Pi and peripherals
- SD Card (64GB) – For Raspberry Pi OS and data storage
- Servo motor - Controls barrier/gate mechanism for vehicle access
- Jumper Wires, Breadboard, Connectors – Circuit connections and prototyping

Software Requirements:

- Operating Systems used:

- Raspberry Pi OS (32-bit or 64-bit) is used.
- Anaconda Navigator installed on a laptop or desktop for code development, testing, and deployment.

- Python Version: Python 3.x (recommended: Python 3.8 or above)
- Database: SQLite (local and lightweight)

VIII. RESEARCH METHODOLOGY

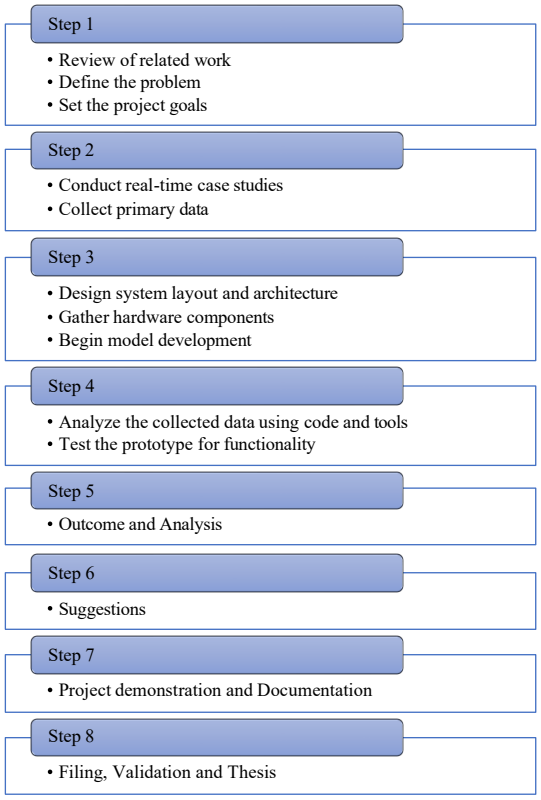

Figure1.2: Research Methodology

IX.BLOCK DIAGRAM

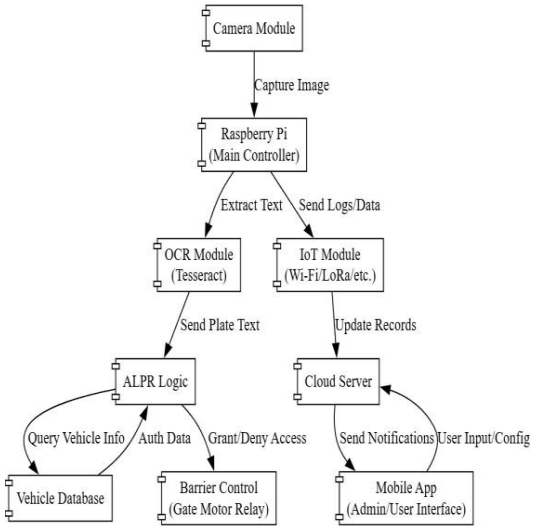

Figure1.3: Block Diagram

The proposed vehicle authorization system uses a layered architecture that integrates image processing, edge computing, IoT communication, and cloud-based monitoring. It captures images of approaching vehicles and uses a Raspberry Pi for processing.

The system uses OCR to read the alphanumeric details from the vehicle's license plate, which are then analysed through an ALPR (Automatic License Plate Recognition) process for identification and verification. records event logs and transmits this to a cloud server for real-time monitoring. The system enhances security, reduces manual intervention, and provides real-time oversight for applications like smart parking systems.

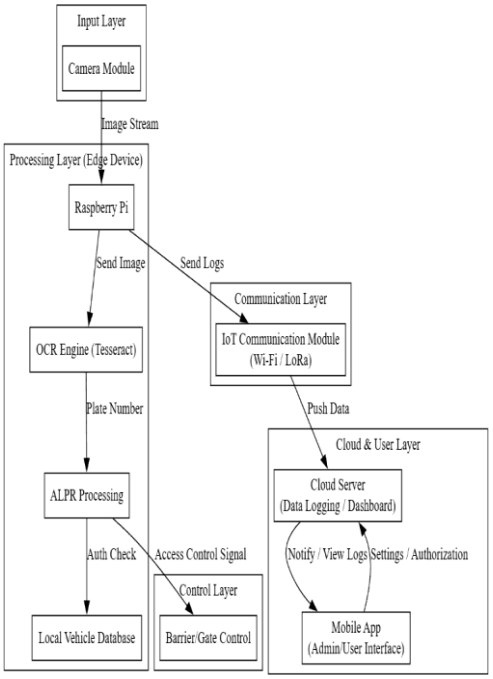

Figure1.4: System Architecture

The system architecture consists of four main layers: Input Layer, Processing Layer, Communication Layer, and Cloud & User Layer, working together to automate vehicle recognition and access control.

1. Input Layer

- The process begins with a Camera Module mounted at the entry or exit point.
- It continuously captures an image stream of incoming vehicles.

2. Processing Layer (Edge Device)

- The image stream is sent to the Raspberry Pi, which acts as the central processing unit.
- The captured image is forwarded to the

OCR Engine (Tesseract), which processes the image and extracts the vehicle's number plate using Optical Character Recognition.

- In this step, the system performs an authorization check by comparing the number plate with the entries in the Local Vehicle Database.
- If the number plate matches an authorized entry, an access control signal is generated.

3. Control Layer

- The access control signal is sent to the Barrier/Gate Control Unit, which either opens or denies access based on the result of the authentication check.

4. Communication Layer

- Simultaneously, the system sends logs (including plate number, timestamp, and status) via the IoT Communication Module (using Wi-Fi or LoRa) to the Cloud & User Layer.

5. Cloud & User Layer

The Cloud Server stores the data for real-time logging, monitoring, and dashboard visualization.

A Mobile App (Admin/User Interface) is used to:

- Notify the vehicle owner of arrival or departure.
- View logs and history of all vehicle entries/exits.
- Manage settings, such as adding/removing authorized vehicles.
- Control authorization status manually if needed.

X.UML DIAGRAMS

Sequence Diagram

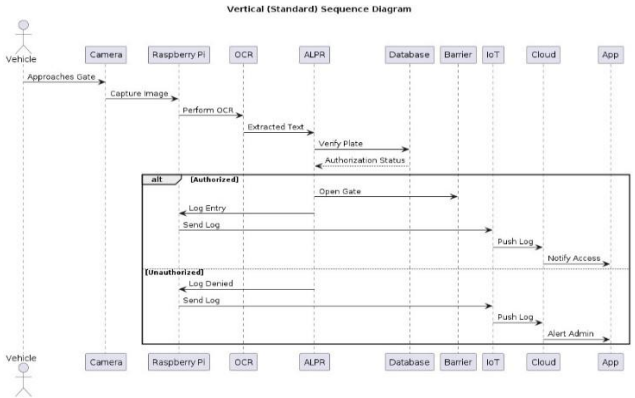

Figure1.5: Sequence Diagram

Complementing the activity flow, the sequence diagram details the time-ordered interactions between actors (the vehicle and admin) and system components like the Camera, Raspberry Pi, OCR, ALPR logic, Vehicle Database, Barrier Control, IoT module, Cloud Server, and Mobile App. It shows the exact message exchanges and responses, such as sending the captured image, processing the OCR, querying the database, triggering the barrier, and logging access. This diagram helps visualize the real-time communications necessary for system operation.

Activity Diagram

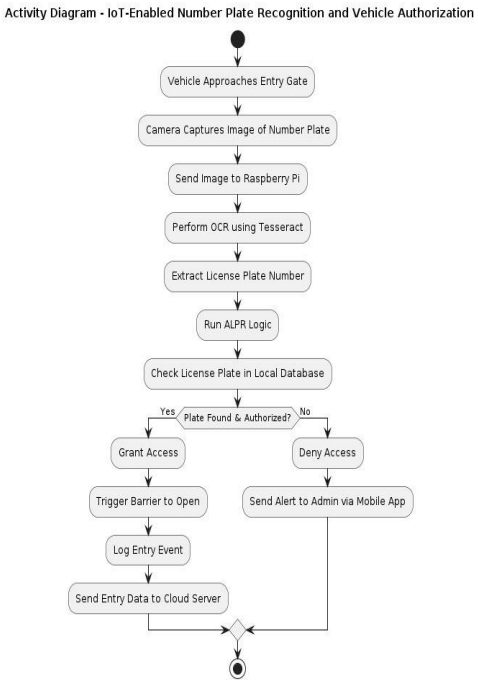

Figure1.6: Activity Diagram

The activity diagram models the dynamic workflow that unfolds as a vehicle approaches the gate. It captures the sequence of activities including image capture, OCR processing, plate recognition, and authorization decision-making. Based on the decision, the gate either opens to allow access or denies entry, followed by logging the event and notifying the system administrator. This diagram clearly portrays the conditional paths and the overall operational process from start to finish.

Class Diagram

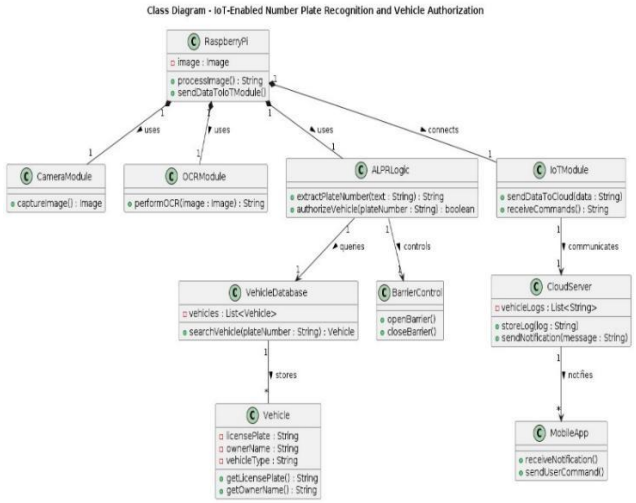

Figure1.7: Class Diagram

The class diagram defines the static structure of the system by representing key classes with their attributes and methods. Classes such as Vehicle, CameraModule, RaspberryPi, OCRModule, ALPRLogic, VehicleDatabase, BarrierControl, IoTModule, CloudServer, and MobileApp are shown with their main properties and behaviors. Relationships between classes — like how the Raspberry Pi uses the OCR and ALPR modules and how the ALPRLogic interacts with the Vehicle Database and Barrier Control — are also depicted. This diagram serves as a foundation for software design and development, mapping out the data model and interactions.

Raspberry Pi Setup Environment

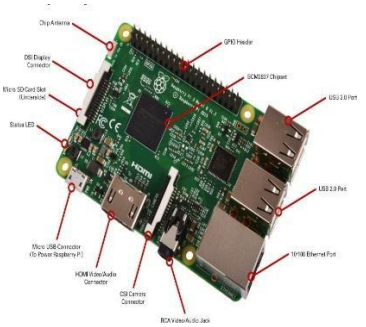

Figure1.8: Raspberry Pi Module

This includes configuring Raspberry Pi OS, developing code for image capture, preprocessing, feature extraction, and classification, integrating components, combining software modules, conducting real-time testing, debugging, and optimizing the user interface.

Raspberry Pi

Model: A Raspberry Pi 4 Model B is commonly used for its enhanced processing power and memory capacity.

Role: Acts as the main processing unit for running the traffic sign recognition algorithms and interfacing with other hardware components.

Camera Module

A Raspberry Pi Camera Module (e.g., v2.1 or compatible USB camera).

Resolution: Typically offers 1080p resolution for high-quality image capture.

Role: Captures real-time video or images of the road environment, providing input for the recognition system.

Figure1.9: Camera Module

Power Supply

Specification: A suitable power supply (5V, 3A) to ensure stable operation of the Raspberry Pi and camera module.

Role: Powers the entire system, ensuring reliable operation in various conditions

Optional Sensors

Ultrasonic Sensors or Lidar: For additional environmental awareness, providing distance measurements to detect nearby objects.

Role: Enhances the system's capability by providing supplementary data for decision-making processes in autonomous applications.

XI.DATASET DESCRIPTION

The graphical data model illustrates the complete data pipeline involved in the vehicle recognition and authorization process using Raspberry Pi. The system starts by collecting various types of input data such as vehicle images or videos, OCR-extracted plate data, authorized vehicle database records, sensor inputs from IoT devices like gates and barriers, GPS and location data, admin inputs (for managing vehicle

permissions), and even external APIs (such as blacklists or alert systems). All these diverse sources feed into a central Data Preprocessing module, which is responsible for cleaning, normalizing, and interpreting the incoming data. For instance, the OCR module may extract alphanumeric characters from a license plate, which then needs to be validated and formatted before further use.

XII.RESULTS AND DISCUSSION

Photo1: Live Detection

Photo2: Live work Detection and information Page

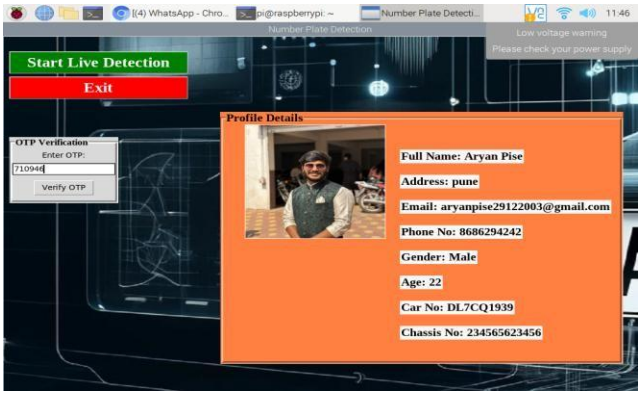

Photo3: OTP Generation

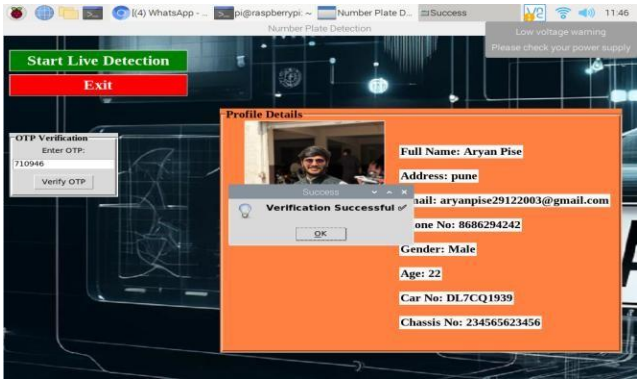

Photo4: OTP Verification Successful Page

The graphical data model illustrates Raspberry Pi's vehicle recognition and authorization process, utilizing input data from various sources, processed by a central Data Preprocessing module, and monitored through real-time operations.

- The system accurately detected and extracted vehicle number plates using OCR.
- Detected number plates were matched with a pre- authorized database.
- On a successful match, an OTP was automatically generated and sent to the registered email.
- The gate was programmed to open only after correct OTP verification, ensuring secure access.
- The system proved to be cost effective, efficient, and suitable for real-time vehicle

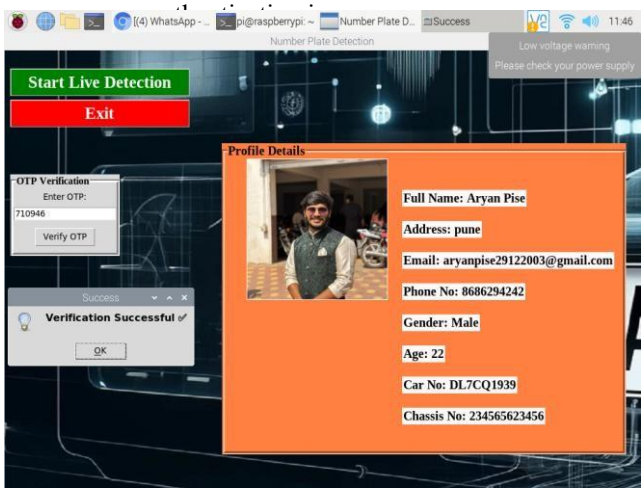

Photo5: Verification Successful Page

XIII. CONCLUSION

The Vehicle Authorization System offers an efficient and affordable method to automate vehicle detection using OCR, Raspberry Pi, and OTP-based

verification. It captures and processes license plates in real-time, helping boost security across various use cases such as gated entries, parking areas, toll booths, and traffic control systems.

By integrating the OTP feature, the system adds an extra layer of security, ensuring that only authorized vehicles and users are granted access. Real-time email notifications further improve monitoring and alerting capabilities, making the system reliable for high-security areas.

The system demonstrates scalability and ease of deployment, making it suitable for a wide range of environments such as gated communities, corporate campuses, and law enforcement applications. With further refinements in image processing and database management, the system can be even more efficient and adaptable for future smart city initiatives and advanced vehicle security systems.

In conclusion, this project showcases the practical applications of affordable technologies like Raspberry Pi and OCR to solve real- world problems in vehicle identification and security, contributing to smarter, safer, and more secure infrastructures.

XIV. FUTURE SCOPE

- ✓ Integration with cloud-based AI for more accurate number plate recognition under poor lighting or angles
- ✓ Expansion to support multi-language number plates and international formats
- ✓ Incorporation of vehicle make, model, and color recognition for enhanced profiling
- ✓ Real-time integration with law enforcement databases for stolen vehicle alerts
- ✓ Use of blockchain technology for secure and tamper- proof vehicle logs
- ✓ Development of a centralized dashboard for city-wide traffic control
- ✓ Compatibility with voice assistants or smart home systems
- ✓ Use of solar-powered IoT devices for energy-efficient deployment in remote areas

REFERENCES

[1]. Allah Ditta et al., "Number plate recognition smart parking management system using IoT" Measurement: Sensors, Elsevier, Volume 37, 2025.

[2]. Prof. Aparna Khairkar et al., "Enhanced Security Mechanism Based On Automated Vehicle Identification" IJCRT, Volume 13, Issue 4, 2025.

[3]. Twana Mustafa Et Al., "Real Time Car Model and Plate Detection System by Using Deep Learning Architectures" IEEE Access Volume 12, 2024.

[4]. Dr. Laya Tojo et al., "Fingerprint Authentication System for Vehicles Using Arduino" JETIR, Volume 11, Issue 5 2024.

[5]. Sanket S. Tabhane et al., "Moving Vehicle Registration Plate Detection System Using OCR" IRJET, Volume: 11 Issue: 04, 2024.

[6]. M.A. Jawale et al., "Implementation of number plate detection system for vehicle registration using IOT and recognition using CNN" Measurement: Sensors, Elsevier, Volume 27, 2023.

[7]. Abdullah Mishal et al., "Advanced Motorbike Ignition System Using Biometric Scanner or RFID Reader" IJIRSET, Volume 12, Special Issue 4, 2023.

[8]. Prof. Bireshwar Ganguly et al., "Research Paper on Vehicle Number Plate Identification Web Application." IJARSCT, Volume 3, Issue 12, 2023.

[9]. Abhilash AS et al., "Biometric Based Authentication for Vehicle Ignition System" IARJSET, Vol. 10, Issue 4, 2023.

[10]. Prof. Bireshwar Ganguly1 et al., "Research Paper on Vehicle Number Plate Identification Web Application" IJARSCT Volume 3, Issue 12, 2023.

[11]. Mrs. Suruchi Dedgaonkar et al., "Automatic number plate recognition" IJNRD Volume 7, Issue 12, 2022.

[12]. Deepak Kumar et al., "Anti-terrorism Vehicle, Authorization System" IJAEM, Volume 4, Issue 5, 2022.

[13]. Prof. Sindhu A S et al., "Vehicle and License Authentication Using Rfid and Finger Print" Ijarcce, Vol. 10, Issue 7, 2021.

[14]. Roman Rak et al., "Digital vehicle identity – Digital VIN in forensic and technical practice" Forensic Science International: Digital Investigation, Elsevier, Volume 39, 2021.

[15]. Sneha S B et al., "Automotive Driver Monitoring System" IJERT, Vol. 10 Issue 01, 2021.

[16]. Aromal A, Brinda H et al., "Theft Vehicle Detection using Automatic License Plate Recognition" IJERT. Volume 9, Issue 13,2021.

[17]. Prof. P. A. More et al., "Biometric Vehicle Access and Ignition System Using Fingerprint Recognition" IJARSCT Volume 6, Issue 1, 2021.

[18]. S. Gayathri et al., "Smart License-Based Vehicle

Safety and Security System" Contemporary Engineering Sciences and Emerging Technologies Volume 2 Issue 1 2021.

[19]. Neha Soni et al., "Number Plate Detection System Using Machine Learning" IJAEM Volume 3, issue 2021.

[20]. Prasun Gokhlani et al., "Automatic Vehicle Identification by Plate Recognition" IRJET Volume: 08 Issue: 07 2021.