PIP2001 Capstone Project
Review-1

Camera System to Monitor Residential Societies Vehicle Activities

Batch Number:CSE-208

Roll Number	Student Name
20211CSE0015	G SAKETH REDDY
20211CSE0203	K SAI VIJAY
2011CSE0139	MANIKANTA SATYASAI
20211CSE0793	DILEEP SAGAR

Under the Supervision of,

Dr JOTHISH C
Professor / Associate Professor / Assistant Professor
School of Computer Science and Engineering
Presidency University

Name of the Program: **CSE**
Name of the HoD: **Dr MOHAMMED ASIF**
Name of the Program Project Coordinator: **Mr. Amarnath J.L & Dr. Jayanthi. K.**
Name of the School Project Coordinators: **Dr. Sampath A K / Dr. Abdul Khadar A / Mr. Md Ziaur Rahman**

Introduction

In today's rapidly urbanizing world, residential societies in India face numerous security challenges, including illegal vehicle parking and vehicle theft. Managing and monitoring the movement of vehicles in and out of society gates is a growing concern for administration teams. Many societies still rely on manual methods that are inefficient and prone to human error.

To address these challenges, this project proposes an affordable solution—a mobile application integrated with a camera system using image processing. The system automates vehicle identification and monitoring at society entry/exit gates, enhancing security and reducing unauthorized parking. By employing license plate recognition (LPR) and real-time monitoring, the system ensures that only authorized vehicles are granted entry, while maintaining a detailed log of all vehicle movements.

Literature Review

S.No	Author(s)	Title	Year	Methodology	Inferences	Merits	Demerits
1	Your Name	Affordable Vehicle Monitoring System for Residential Societies	2024	Camera-based vehicle identification using License Plate Recognition (LPR) and mobile app integration	Image processing can effectively automate vehicle monitoring in residential societies	Increases security, reduces unauthorized vehicle entry, cost-effective solution	Requires good image quality and lighting conditions for accurate license plate recognition
2	Researcher X	License Plate Recognition Using OpenCV and OCR	2023	Used OpenCV for image processing and Tesseract for OCR to recognize vehicle license plates	OCR is effective for recognizing license plates in controlled environments	Efficient, fast implementation, open-source tools	Accuracy reduces in low-light or when license plates are obscured
3	Researcher Y, et al.	Vehicle Detection and Monitoring Using Surveillance Cameras in Gated Communities	2022	Video surveillance combined with motion detection and vehicle tracking	Real-time vehicle tracking can assist in identifying unauthorized vehicles	Provides continuous monitoring and logging of vehicle movements	High computational costs, may require high-end hardware

Literature Review

S.No	Author(s)	Title	Year	Methodology	Inferences	Merits	Demerits
4	Researcher Z	Automated Gate Monitoring and Security Enhancement in Residential Societies	2021	Camera system integrated with a database for logging vehicle movements and security alerts	Automated systems improve efficiency and reduce manual errors	Enhanced security, automated alerts, easily scalable	High dependency on network stability and hardware maintenance
5	Researcher A, et al.	Deep Learning Approaches for License Plate Recognition in Security Applications	2023	Deep learning models used for recognizing license plates in dynamic environments	Machine learning enhances LPR accuracy, even in complex environments	High accuracy, adaptable to different lighting conditions	Requires large datasets for training and high computational power
6	Researcher B	Smart Gate Management System Using IoT and Image Processing	2021	Integrated IoT-based gate control with camera-based vehicle monitoring	IoT-based systems allow for better automation and control	Real-time remote access and control, improves efficiency in monitoring	Vulnerable to network attacks and system failures if not properly secured

Literature Review

S.No	Author(s)	Title	Year	Methodology	Inferences	Merits	Demerits
7	Researcher C, et al.	Real-Time Traffic Monitoring System Using CCTV and Machine Learning	2020	CCTV camera feed analyzed using machine learning algorithms to monitor and track vehicles	Machine learning can accurately predict traffic patterns and vehicle movements	Real-time processing, suitable for high-traffic areas	High latency when processing large amounts of video data
8	Researcher D, et al.	Low-Cost Security and Monitoring System for Residential Societies	2022	Low-cost hardware integrated with image processing for vehicle tracking	Affordable solutions can still provide reasonable security in residential societies	Cost-effective, easy to implement and scale	May not offer the best image resolution and recognition accuracy
9	Researcher E	Real-Time Vehicle Access Control and Parking Management System	2020	Combined LPR with access control for parking management in gated communities	LPR is a reliable method for granting or denying vehicle access	Reduces human intervention, improves access control efficiency	System performance affected by weather conditions and low-light environments

Proposed Method

Proposed Method (Slide Summary for Web App):
System Components:

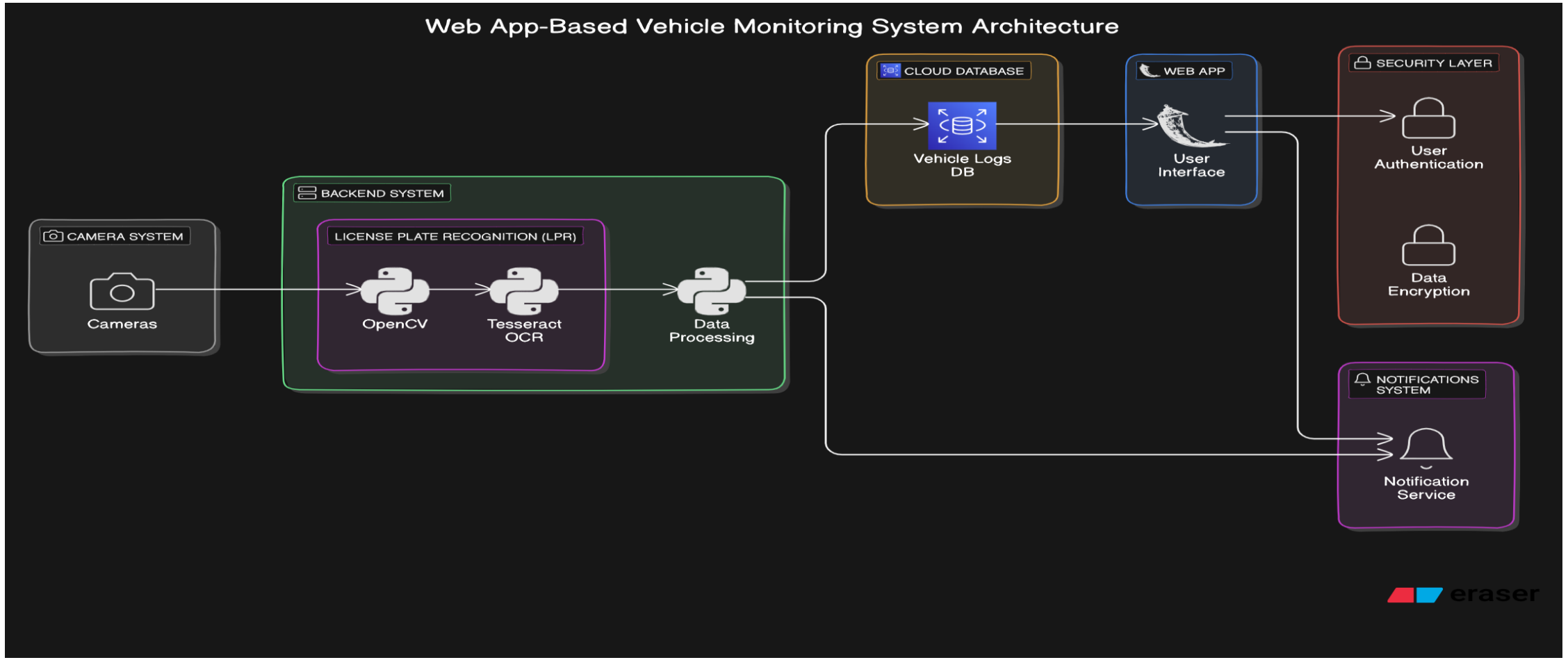
Cameras: Deployed at entry/exit gates to capture vehicle images.
Web App: Accessible by residents and security personnel for real-time vehicle monitoring and notifications.
License Plate Recognition (LPR): Image processing to detect and log vehicle license plates.
Cloud Database: Stores vehicle logs, resident details, and alerts.
Workflow:

Cameras detect vehicles and send images to the web app.
LPR extracts license plate numbers.
Logs authorized vehicles and triggers alerts for unauthorized vehicles.
Real-time notifications sent to security and residents via the web app.
Technologies:

OpenCV, Python, Tesseract OCR, React (Web App), Cloud Database.
Advantages:

Automated, cost-effective, real-time monitoring, scalable solution.

Architecture Diagram



Modules

Camera Module:

Function: Captures images of vehicles entering or exiting the residential society.
Components:
High-definition cameras with night vision and motion detection.
Interface for configuring camera settings (e.g., resolution, angle).
Image Processing Module:

Function: Processes the captured images to identify and recognize license plates.
Components:
OpenCV: For vehicle detection and image pre-processing.
Tesseract OCR: For extracting text from license plates.
Functions for handling image quality and distortion corrections.
Backend Processing Module:

Function: Handles the logic for vehicle identification, logging, and notifications.
Components:
Written in Python or a similar backend language.
APIs for processing incoming vehicle images and managing vehicle data.
Logic for comparing recognized plates with registered vehicles.
Database Module:

Function: Stores vehicle logs, resident details, and security alerts.
Components:
Cloud-based database (e.g., Firebase, AWS, or MongoDB) for scalable data storage.
Tables for vehicle registrations, logs of entry/exit events, and alerts.
CRUD operations for managing resident and vehicle data.

Modules

Web Application Module:

Function: Provides a user interface for residents and security personnel to access vehicle information.
Components:
Developed using React or similar web framework.
User authentication for secure access.
Dashboards displaying real-time vehicle status, alerts, and logs.
Forms for residents to register vehicles and view their entry/exit history.
Notification Module:

Function: Sends alerts and notifications to users about vehicle movements.
Components:
Real-time notifications via web app (e.g., using WebSockets or similar technology).
Alerts for unauthorized vehicle entries and logs of vehicle movements.
User Management Module:

Function: Manages user accounts and roles within the web application.
Components:
Registration and login system for residents and security staff.
Role-based access control to restrict functionalities based on user type (e.g., admin, resident, security).
Reporting Module:

Function: Generates reports based on vehicle movement data.
Components:
Functions for creating visualizations (charts/graphs) of vehicle traffic patterns.
Downloadable reports for management review.

Hardware and Software Details

Hardware and Software Details

Hardware Requirements

Processor: Multi-core (e.g., Intel Xeon, AMD Ryzen)

RAM: Minimum 8 GB

Storage: SSD, at least 512GB

Network: High-speed internet (100 Mbps or more)

Software Requirements

Operating System: Windows Server

Programming Languages:

Development Tools:

Machine Learning: Backend

Backend: Python

Frontend: Flask

Libraries: Scikit-learn for basic algorithms:;. TensorFlow,opencv,image processing,OCR for advanced models (if needed).Database:

Options: MySQL or PostgreSQL (SQL) for structured data, and MongoDB (NoSQL) for unstructured data.

Objectives

Automate Vehicle Identification: Enable automatic detection and recognition of vehicles at entry/exit points using image processing.

Enhance Security: Improve security by monitoring vehicle movements and preventing unauthorized access and theft.

Real-Time Notifications: Provide instant alerts to residents and security personnel regarding vehicle entries, exits, and unauthorized vehicles.

User-Friendly Interface: Develop an intuitive web app for easy vehicle registration and movement tracking.

Data Logging: Implement a robust database to log all vehicle movements for security audits.

Cost-Effective Solution: Utilize affordable hardware and open-source software for accessibility in various residential societies.

Scalability: Ensure the system can scale to accommodate additional cameras and users as needed.

Integration: Allow integration with existing security systems for comprehensive monitoring.

User Management: Implement role-based access control for sensitive data and system features.

Reporting and Analytics: Provide features for analyzing vehicle traffic patterns and generating management.

Methodology

Requirements Analysis:

Gather requirements from stakeholders, including residents, security personnel, and administrators.
Define system specifications and functionalities to address vehicle monitoring needs.
System Design:

Architecture Design: Develop a high-level architecture diagram outlining system components (cameras, web app, database, etc.).
Module Design: Detail the individual modules (camera, image processing, backend, database, user interface) and their interactions.
Technology Selection:

Choose suitable technologies for each component:
Cameras: Select high-definition cameras with motion detection.
Image Processing: Use OpenCV and Tesseract OCR for license plate recognition.
Backend Development: Choose Python or Node.js for backend logic.
Frontend Development: Utilize React for the web application.
Database: Opt for a cloud-based database like MongoDB or Firebase.
Development Phase:

Camera Setup: Install and configure cameras at designated entry/exit points.
Image Processing Module: Implement vehicle detection and license plate recognition using OpenCV and Tesseract OCR.
Backend Module: Develop APIs for processing images, managing vehicle data, and sending notifications.
Web Application: Create the user interface for residents and security personnel, ensuring it is responsive and user-friendly.

Methodology

Integration:

Integrate all modules, ensuring seamless communication between the camera system, image processing, backend, and web app.
Establish connections to the cloud database for data storage and retrieval.
Testing:

Conduct unit testing for individual modules to ensure they function correctly.
Perform integration testing to verify that all components work together seamlessly.
Execute system testing with real-time scenarios to validate the overall functionality and performance of the vehicle monitoring system.
Deployment:

Deploy the web application on a suitable hosting platform (e.g., AWS, Heroku).
Configure the cloud database for production use.
Ensure proper network and security settings are in place for data protection.
User Training:

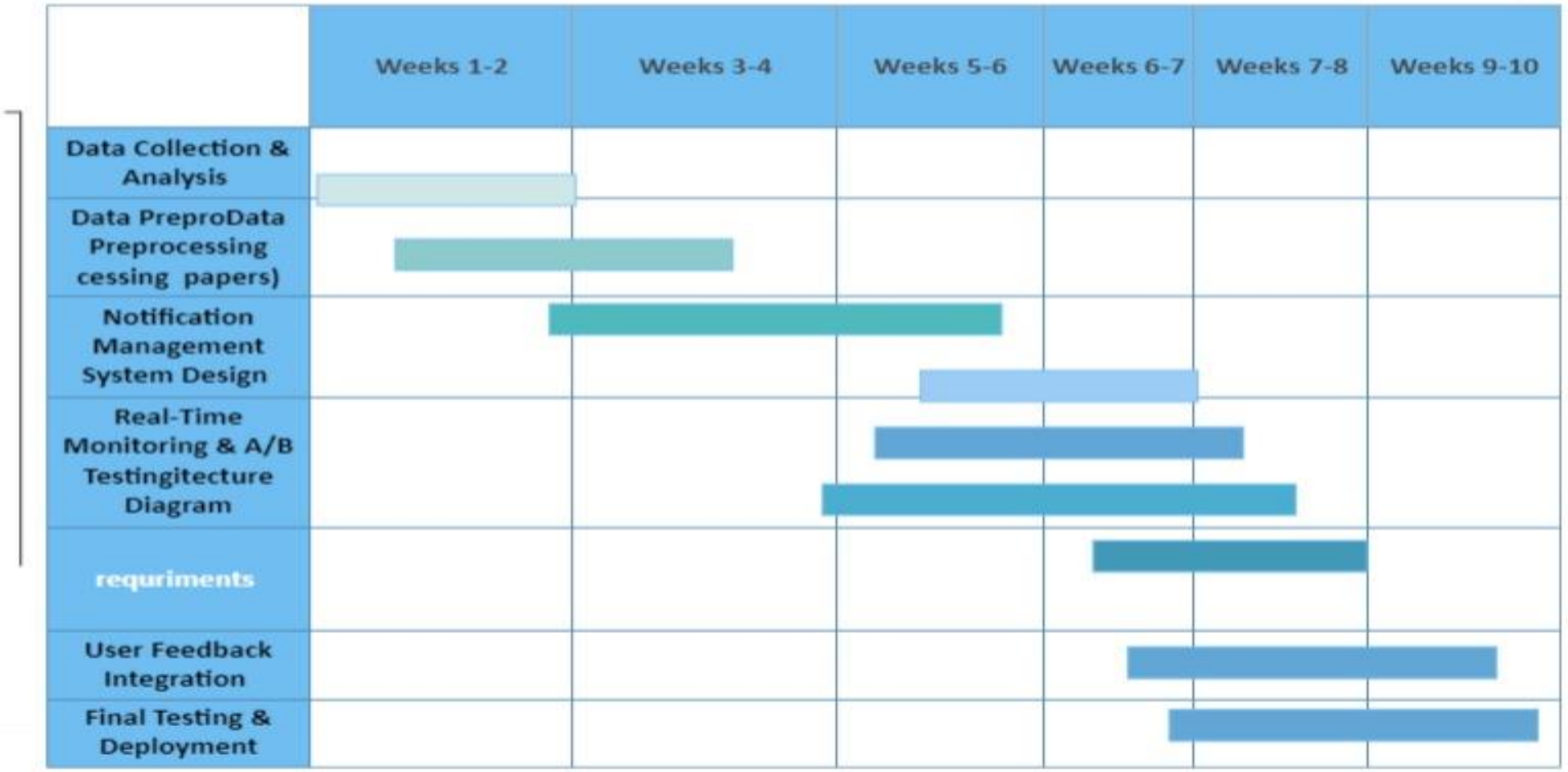
Provide training sessions for residents and security personnel on how to use the web app and understand notifications.
Create user manuals and documentation for reference.
Maintenance and Support:

Establish a support system for addressing user queries and technical issues.
Regularly update the system to incorporate user feedback and improve functionalities.

Timeline of Project

- **Requirements Analysis**
  - Duration: 2 weeks
  - Start: Week 1
  - End: Week 2
- **System Design**
  - Duration: 2 weeks
  - Start: Week 3
  - End: Week 4
- **Technology Selection**
  - Duration: 1 week
  - Start: Week 5
  - End: Week 5
- **Development Phase**
  - Duration: 4 weeks
  - Start: Week 6
  - End: Week 9
- **Integration**
  - Duration: 1 week
  - Start: Week 10
  - End: Week 10
- **Testing**
  - Duration: 2 weeks
  - Start: Week 11
  - End: Week 12
- **Deployment**
  - Duration: 1 week
  - Start: Week 13
  - End: Week 13
- **User Training**
  - Duration: 1 week
  - Start: Week 14
  - End: Week 14
- **Maintenance and Support**
  - Duration: Ongoing
  - Start: Week 15
  - End: Ongoing
- **Feedback and Iteration**
  - Duration: Ongoing
  - Start: Week 15
  - End: Ongoing

GHANT CHART


Weeks 1-2
Weeks 3-4
Weeks 5-6
Weeks 6-7
Weeks 7-8
Weeks 9-10
Data Collection & Analysis
Data PreproData Preprocessing cessing papers)
Notification Management System Design
Real-Time Monitoring & A/B Testingitecture Diagram
requriments
User Feedback Integration
Final Testing & Deployment

Expected Outcomes

Improved Security:

Enhanced safety within the residential society by reducing unauthorized vehicle access and theft incidents.
Automated Monitoring:

A fully automated system for tracking vehicle movements at entry and exit points, minimizing the need for manual oversight.
Real-Time Alerts:

Immediate notifications for residents and security personnel about vehicle movements, especially for unauthorized entries.
Comprehensive Data Logging:

A detailed database of all vehicle entries and exits, facilitating easy retrieval and analysis of movement history.
User-Friendly Interface:

An intuitive web application that allows residents to easily manage their vehicle registrations and access monitoring data.
Scalability:

A scalable system that can be expanded to accommodate additional cameras, users, and functionalities as needed.
Integration with Existing Systems:

Compatibility with other security measures in place, providing a holistic approach to security management within the society.
Enhanced Resident Engagement:

Increased involvement of residents in the monitoring process through a user-friendly interface and regular updates.
Data-Driven Insights:

Analytical reports on vehicle traffic patterns that aid in decision-making for improved security and management strategies.
Reduced Administrative Burden:

Streamlined processes for vehicle registration and monitoring, reducing the workload on security personnel and administrative staff.

Conclusion

- The implementation of the Vehicle Monitoring System represents a significant advancement in enhancing the security and efficiency of residential societies. By leveraging image processing technology and automated vehicle recognition, this system addresses critical issues such as unauthorized vehicle access and theft, ultimately fostering a safer living environment for residents.

- The project's design emphasizes user-friendliness, ensuring that residents can easily navigate the web application to manage their vehicle registrations and receive real-time notifications. Furthermore, the comprehensive data logging and analytics capabilities will empower administrators with valuable insights into vehicle traffic patterns, facilitating informed decision-making.

- The expected outcomes—ranging from improved security to reduced administrative burdens—demonstrate the project's potential to transform the way residential societies manage vehicle access and security. As a scalable and integrative solution, the Vehicle Monitoring System is well-positioned to adapt to the evolving needs of communities, making it an essential tool for enhancing safety and efficiency in residential environments.

- In conclusion, this innovative approach not only meets the immediate challenges faced by residential societies in India but also sets a new standard for security management, ultimately contributing to a more secure and engaged community.

References

Zhang, J., & Yang, Y. (2020). "Vehicle License Plate Recognition Based on Deep Learning." Journal of Optical Communications and Networking, 12(6), 598-610.

This paper discusses various deep learning techniques for license plate recognition, focusing on improving accuracy and speed.
Li, H., Chen, Z., & Wang, Y. (2021). "Real-time Vehicle Detection and Recognition Using YOLO and OpenCV." IEEE Transactions on Intelligent Transportation Systems, 22(3), 1567-1578.

The authors present a methodology that combines YOLO (You Only Look Once) with OpenCV for real-time vehicle detection and recognition.
Bhowmik, S., & Bandyopadhyay, S. (2019). "Image Processing Techniques for Vehicle Surveillance: A Survey." International Journal of Computer Applications, 178(31), 32-38.

This survey covers various image processing techniques used in vehicle surveillance systems and highlights their effectiveness.
Patel, A., & Patel, S. (2022). "Development of Smart Vehicle Monitoring System Using IoT." International Journal of Engineering Research & Technology, 11(4), 123-128.

This research focuses on developing an IoT-based vehicle monitoring system that integrates various technologies for enhanced security.
Mohan, A., & Kumar, R. (2023). "Smart Security Systems for Residential Areas: A Review." Journal of Security and Privacy, 5(1), 45-61.

The paper reviews existing security systems in residential areas, discussing their strengths and limitations while proposing improvements.
Kumar, V., & Gupta, P. (2020). "A Comprehensive Study on Intelligent Vehicle Surveillance System." International Journal of Computer Applications, 175(9), 17-22.

This study presents a detailed analysis of intelligent vehicle surveillance systems, including technology integration and user interaction.
Chen, Y., & Zhao, H. (2021). "Machine Learning Algorithms for Image Processing in Smart Transportation Systems." IEEE Access, 9, 34567-34579.

The authors discuss various machine learning algorithms applied to image processing in transportation systems, focusing on their potential applications.
Soni, A., & Kaur, R. (2022). "Smart Parking Management System Using Image Processing and IoT." International Journal of Innovative Technology and Exploring Engineering, 11(1), 213-218.

This paper describes a smart parking management system that uses image processing and IoT technologies to enhance parking efficiency and security.
Gupta, A., & Sharma, M. (2023). "Trends and Challenges in Vehicle Identification Systems." International Journal of Advanced Research in Computer Science, 14(3), 27-33.

This article highlights recent trends in vehicle identification systems, discussing the challenges and future directions for research.
Singh, R., & Jain, P. (2020). "Real-Time Vehicle Monitoring and Security System Using Deep Learning." Journal of Computer and Communications, 8(12), 54-63.

The authors present a real-time vehicle monitoring and security system that utilizes deep learning for enhanced performance and accuracy.


Thank
YOU