International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

A Review on Smart Car Parking and Intrusion Detection System (Astra Vision)

Alok Ahirrao[1], Dhanashri Mirajkar[2], Gauri Raykar[3] , Tanaya Varpe[4]

[1] *Department of Artificial Intelligence and Machine Learning ,Progressive Education Society Modern College of Engineering,Pune,India.*

[2] *Department of Artificial Intelligence and Machine Learning ,Progressive Education Society Modern College of Engineering,Pune,India.*

[3] *Department of Artificial Intelligence and Machine Learning ,Progressive Education Society Modern College of Engineering,Pune,India.*

[4] *Department of Artificial Intelligence and Machine Learning ,Progressive Education Society Modern College of Engineering,Pune,India.*

---***---

Abstract -

Urban parking systems face significant challenges in efficiency and security. Traditional methods often result in traffic congestion, delays in finding parking spots, and increased vulnerability to theft, vandalism, and unauthorized access, emphasizing the need for effective management and surveillance. Smart parking systems using technologies like computer vision offer solutions but struggle with accuracy under poor lighting, weather conditions, and blocked views. High computational demands for real-time processing further limit their scalability. Real-time parking allocation adds complexities such as managing large vehicle volumes, ensuring fairness, and aligning with user preferences in urban settings. Security systems, while essential, face issues like high false alarms, adaptation to new threats, and privacy concerns. Despite advancements, smart parking systems remain limited by environmental challenges, computational requirements, and the need for user-focused, secure designs. Innovations are crucial to create adaptive, efficient systems.

***Key Words*: . Smart parking systems, computer vision, computational requirements, real-time processing, security systems**

1.INTRODUCTION

ASTRA VISION is an advanced computer vision-based solution designed to tackle challenges in urban parking management and security. Integrating cutting-edge technologies in image processing, artificial intelligence, and real-time monitoring, the system offers a unified approach to efficient parking allocation and enhanced safety.

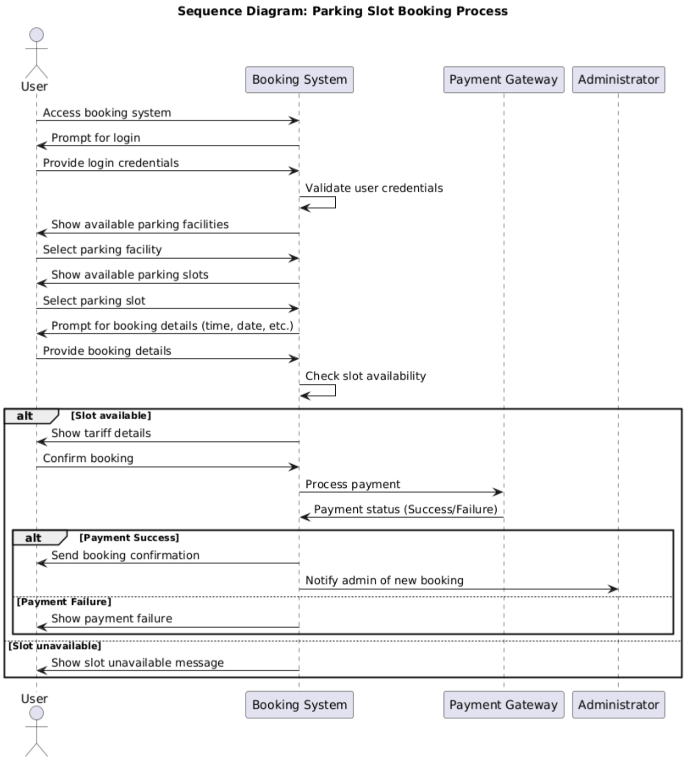
Through its computer vision capabilities, ASTRA VISION automates parking space detection, monitors slot availability, and strengthens security through intelligent surveillance. This innovative system minimizes inefficiencies, alleviates user frustrations, and proactively addresses security concerns, making it a transformative tool for urban parking management.

Urban areas face growing parking challenges due to increasing congestion, leading to lost time, traffic delays, and user dissatisfaction caused by inadequate guidance and limited parking availability. Security issues, including unauthorized access, theft, and vandalism, further complicate parking management, with traditional surveillance systems often falling short in mitigating these risks.

ASTRA VISION was developed to address these challenges by integrating modern computer vision techniques into a scalable and user-centric system. The project aims to optimize space usage, improve user experiences, and enhance security measures while being adaptable to various urban environments.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 01 | Jan - 2025 SJIF Rating: 8.448 ISSN: 2582-3930

2. SEQUENCE DIAGRAM



3. SOFTWARE REQUIREMENT SPECIFICATION

Functional Requirements:

Development Tools:

1. Code Editor/IDE:

- Visual Studio Code

2. Programming Languages:

- Python
- JavaScript

3. Libraries:

- Numpy and Pandas: For data manipulation and numerical computation in Python.
- TensorFlow: A key library for machine learning and deep learning, used to develop neural networks and deploy models on different platforms.
- Matplotlib and Seaborn: Tools for data visualization. Matplotlib offers control, while Seaborn creates attractive statistical plots.
- Scikit-learn: Provides algorithms for classical machine learning tasks like classification, regression, and clustering.
- Streamlit: Used to quickly build and share web apps for machine learning and data science.

4. Database:

- MongoDB
- MongoDB Compass (GUI for managing MongoDB)

5. Frameworks and Libraries:

- Node.js: JavaScript runtime for backend development.
- Reactx: JavaScript library for building user interfaces (frontend).
- Tailwind CSS: A utility-first CSS framework for styling.
- Express: Backend framework for building APIs.
- FastAPI and Flask: Web frameworks for creating APIs, with FastAPI for asynchronous operations and Flask for simpler applications.

6. Machine Learning Algorithm:

- XGBoost: A powerful algorithm for boosting decision trees, often used with structured or tabular data.

Non-Functional Requirements:

1. Performance Requirements:

- Response Time: The system must process video feeds with a response time under 1 second for real-time event detection (e.g., car parking, intrusion detection).
- Throughput: The system should handle up to 100 concurrent video streams without performance degradation, maintaining at least 15 frames per second (FPS) per stream.
- Resource Usage: The system should optimize CPU and GPU resource use, ensuring efficient load distribution. Memory usage should be kept under 75% of available memory during peak load to avoid bottlenecks.

2. Reliability Requirements:

- Fault Tolerance: The system should gracefully handle
hardware or software failures, with automatic failover to backup systems for continuous operation.
- Recovery Processes: In case of failure, the system must support automatic recovery and

restore operations within 5 minutes. Regular backups of configurations and data should be scheduled.

- Uptime: The system should maintain an uptime of 99.9% to ensure high availability in critical security environments.

3. Usability Requirements:

- User Interface Design: The interface must be intuitive, user-friendly, and easy to navigate. It should provide customizable dashboards and accommodate users with varying technical skills.
- Accessibility: The system should meet accessibility standards (e.g., WCAG 2.1), including features like screen reader support, keyboard navigation, and high-contrast modes.
- Training and Support: The system should offer comprehensive training materials, user documentation, and ongoing technical support to help users effectively operate the system.

Design and Implementation Constraints :

- **Processing Power**: Real-time video processing demands significant computational resources, especially for high-density camera networks.
- **Data Privacy**: The system must adhere to privacy laws, requiring strict data encryption and access control.
- **Integration:** The system must be compatible with existing security infrastructure, which may require custom development or adaptations.
- **Scalability**: The system should efficiently scale to handle increasing numbers of cameras and data sources without degrading performance.
- **Resource Constraints:** Limited budgets or available hardware may affect feature scope or deployment extent.

Assumptions and Dependencies:

- **Availability of High-Quality Video Data:** The system assumes access to high-definition video feeds from well-placed cameras.
- **Network Stability**: Reliable, high-speed internet or intranet connections are necessary for data transmission and remote monitoring.
- **Compliance with Legal Standards:** The system assumes all deployed environments will comply with relevant legal and regulatory standards for video surveillance and data retention.
- **Integration with Existing Systems:** The system assumes minimal changes are needed to integrate with existing security infrastructure.

Requirement Analysis Techniques:

Workshops/Focus Groups:

- Collaborative sessions with stakeholders to gather insights and discuss requirements in a structured environment.
- Pros: Engages multiple stakeholders at once, allows for real-time clarification.
- Cons: Can be time-consuming and may lead to groupthink.

Observation:

- Observing users or stakeholders performing tasks to understand how the system will be used in real-world conditions.
- Pros: Provides practical insights and highlights user needs in context.
- Cons: Limited to observing current behavior; may miss underlying needs or future requirements.

Prototyping:

- Developing an early version of the system or components to gather feedback and refine requirements.
- Pros: Allows users to interact with a working model, clarifying needs and expectations.
- Cons: Can be time-consuming and may lead to misunderstandings if not well-managed.

4. RELATED WORK

Advancements in smart surveillance and parking management systems have been explored extensively in recent years to address challenges related to urban congestion and security.

Conventional parking systems often rely on manual processes or basic sensors, leading to inefficiencies in space utilization and increased user dissatisfaction. With the advent of computer vision and machine learning, automated systems have emerged, offering real-time detection and monitoring of parking spaces and vehicles.

These systems integrate deep learning models for tasks such as license plate recognition, vehicle counting, and parking violation detection, demonstrating improved efficiency and accuracy.

In addition to parking management, modern intrusion detection systems utilize advanced algorithms to monitor and secure spaces against unauthorized access. Technologies like Faster R-CNN and YOLO have been employed for object detection, while deep learning models specialize in detecting unusual activities, such as fire, smoke, or helmet violations. Despite their progress, these systems often face challenges, including scalability, environmental limitations, and high false alarm rates. By building on these innovations, Astra Vision seeks to overcome these limitations by providing a unified, multi-functional platform that combines robust parking management with advanced security monitoring capabilities.

5. FUTURE SCOPE AND WORK

Future development of Astra Vision can focus on expanding its functionality and adaptability, strengthening its position as an advanced smart parking and security system. Features like personalized user options, such as reserved parking spaces and loyalty benefits, could be introduced through a mobile application to enhance user convenience and control.

1. Augmented Reality Navigation:
   - Develop AR-based navigation tools to guide users to available parking spaces.
   - Improve user experience with real-time and interactive visual directions.
2. Energy-Efficient Solutions
   - Use eco-friendly technologies, such as solar-powered surveillance cameras.
   - Reduce the environmental footprint of the system.
3. Continuous Model Improvement
   - Update machine learning models with new datasets to improve detection accuracy.
   - Adapt to different environments and scenarios for enhanced reliability.
4. Scalable Multi-Language Support
   - Develop multi-language support in the interface for global deployment.
   - Cater to diverse user bases in various regions.
5. Advanced Data Privacy Measures
   - Introduce stronger encryption and data anonymization techniques.
   - Ensure compliance with updated privacy regulations to build user trust.

Integrating Astra Vision with broader smart city infrastructure, such as real-time traffic and public transit systems, could help optimize parking availability and reduce congestion citywide. These advancements would transform Astra Vision into a comprehensive, scalable, and sustainable smart city solution, contributing to more efficient and eco-friendly urban environments.

6. CONCLUSIONS

In summary, Astra Vision provides an innovative approach to tackling the challenges of urban parking management and security. By leveraging advanced computer vision, artificial intelligence, and real-time monitoring, the system delivers a holistic solution for optimizing parking space usage while addressing security risks often found in conventional setups. With features such as automated parking slot detection, dynamic allocation, and effective intrusion detection, Astra Vision improves user convenience, maximizes resource efficiency, and enhances overall safety in parking environments.

While the system is already robust, there is significant potential for future development. Integrating predictive analytics, cloud-based scalability, and augmented reality could further expand its capabilities and adaptability, aligning with the vision of smart cities. Strengthening data privacy measures and incorporating energy-efficient technologies would also contribute to sustainable urban growth. Astra Vision stands as a forward-thinking solution, designed to address the needs of modern urban environments, offering a safer, more efficient, and user-friendly parking experience.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our faculty advisor, Prof. Sapna Bhirud for their continuous support, guidance, and valuable insights throughout the development of this project. Their expertise and encouragement were instrumental in the successful implementation of the Smart Car Parking and Intrusion Detection System (Astra Vision).

We extend our heartfelt thanks to PES Modern College of Engineering for providing us with the resources and facilities that made this project possible. The supportive academic environment fostered by our institution has greatly enhanced our learning experience and motivated us to explore innovative solutions.

REFERENCES

1. M. M. A. Hasnat, M. M. Rahman, and K. Hasan, "Automatic Parking Space Detection and Slot Allocation Method for Efficient Car Parking," 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 2020, pp. 120-125. doi: 10.1109/TENSYMP50017.2020.9230926.

2. J. Rizvi, R. Mehjabeen, and M. M. Uddin, "Intrusion Detection System using Deep Learning and Machine Learning Techniques for Parking Security," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 912-917. doi: 10.1109/ICCMC51019.2021.9418356.

3. D. Zhang, G. Wang, and L. Jiao, "Real-Time Parking Slot Detection for Parking Assistance System in Different Scenarios," 2017 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 2017, pp. 2064-2069. doi: 10.1109/ICMA.2017.8016149.

4. P. Jayaraman, A. Ram, and V. Kandasamy, "Slot Allocation in Smart Parking Systems Using Reinforcement Learning," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2019, pp. 1095-1100. doi: 10.1109/UEMCON47517.2019.8993113.

5. T. Kato, S. Muramatsu, and H. Asai, "Vision-Based Detection of Parking Spaces for Driver Assistance Systems," 2015 IEEE/SICE International Symposium on System Integration (SII), Nagoya, Japan, 2015, pp. 589-594. doi: 10.1109/SII.2015.7405051.