Tribhuvan University

Institute of Science and Technology

A PROJECT-PROPOSAL

REPORT ON

Parking Space Detection System

Submitted to

Department of Bsc.CSIT

Nepalaya College

In partial fulfillment of the requirements for the BSc.CSIT

Submitted by

Supriya Adhikari (26697/077)

Samiksha Ghimire (26690/077)

Pranisha Neupane (26681/077)

Under the Supervision of

Narayan Chalis

1. Introduction

Parking spaces are in short supply in contemporary urban landscapes due to the proliferation of vehicles. There is a considerable amount of congestion in urban areas, which can result in lost time, fuel, and increased environmental impacts. There is often a lack of information provided by traditional parking management systems, which intensify the issue. The project titled "Parking Space Detection System", built using Python, Flask, OpenCV an aims to revolutionize parking space management and access through the intersection of advanced technology and urban planning. Previous vehicle traffic studies have estimated that around 30% of vehicles traveling in inner city areas are made up of drivers searching for a vacant parking space [1].

In cities, parking spaces are becoming scarce as the population density continues to increase and the number of vehicles on the road continues to increase. A major problem in metropolitan areas is searching for parking spaces [2]. Traffic congestion and frustrated drivers are caused by conventional parking infrastructure struggling with the ever-increasing demand. Recognizing the need for a smarter, technology-driven solution, this project proposes the integration of OpenCV, a powerful computer vision library, to process live video feeds from surveillance cameras installed in parking lots. This system uses OpenCV's image processing and pattern recognition capabilities to detect parking spaces accurately and quickly, and to provide the admin with updates through a user-friendly web interface. The interface is accessible only by the admin, ensuring that only authorized personnel can manage and monitor the system [3].

Existing parking management systems often lack the ability to adapt to dynamic parking scenarios and fail to provide drivers with timely information about available spaces. This project aims to bridge this gap by developing an intuitive web interface utilizing Flask, a Python web framework. The interface will facilitate access to information about parking space availability for the admin, allowing for efficient and centralized management of parking spaces. A comprehensive solution to the challenges 2 posed by modern urban parking management has been offered by the system because of its adaptability, and scalability [4].

2. Problem Statement

1. **Historical Challenges in Parking:**
   Finding a vacant parking space in multilevel parking lots has long been a challenging task. This issue becomes even more pronounced during weekends or public holidays when parking demand surges. Research indicates that a significant majority—about 86%—of drivers experience difficulties in locating available parking spots. This historical problem highlights a persistent issue in urban mobility and infrastructure management.

2. **Current Urban Parking Issues:**
   Today, urban areas are grappling with increased parking congestion due to rapid population growth and rising vehicle ownership. Existing parking facilities often suffer from inefficient management practices. This inefficiency results in a chaotic parking experience where drivers face confusion and frustration, and the available parking space is not utilized optimally. As the number of vehicles continues to rise, the inadequacies of current parking systems become more pronounced.

3. **Future-Proofing Parking Management:**
   Looking ahead, the project aims to address these ongoing issues by implementing a Parking Space Detection System. By harnessing computer vision technology, this system will optimize the management of parking resources. It will offer a user-friendly web interface exclusively for administrative use, enabling better oversight and control. Additionally, the system will analyze parking utilization patterns, which will aid in redesigning parking lots for improved efficiency. Automatic notifications will further enhance the system's effectiveness by keeping users informed of parking space availability. This approach is designed to overcome historical challenges, improve current inefficiencies, and pave the way for a more efficient and user-centric future in urban parking management.

3. Objectives

This project aims to develop a parking space detection system for the identification of parking space status. Simultaneously, we plan to design an intuitive web interface with Flask, featuring automatic notifications based on space availability, to enhance user convenience and optimize the overall parking experience in urban areas.

1. **Development of a Parking Space Detection System:**
   The primary objective of this project is to develop a highly efficient parking space detection system that uses OpenCV for precise and swift identification of parking space statuses. By employing advanced computer vision techniques, the system will be capable of accurately distinguishing between vacant and occupied parking spaces. This technological approach aims to address the challenges of locating parking spaces by providing real-time updates on space availability. The focus will be on creating a robust system that ensures high accuracy and reliability, which is crucial for enhancing the overall parking experience and reducing the time drivers spend searching for parking.

2. **Design of an Intuitive Web Interface:**
   Another key objective is to design an intuitive web interface using Flask that complements the parking space detection system. This interface will feature an automatic grid representation system that visually displays the status of each parking space. By integrating real-time data from the detection system, the web interface will provide users with an easy-to-understand layout of parking space availability. Additionally, the system will include automatic notifications to alert users about changes in space availability, further enhancing user convenience. The goal is to create a seamless and user-friendly experience that simplifies parking management and improves the efficiency of parking space utilization in urban areas.

4. Methodology

a. Requirement Identification

i. Study of Existing System / Literature Review

An analysis of the literature is an essential component of a Parking Space Detection System project report. This report presents a comprehensive overview of research and studies that have been carried out related to the topic. During this project, our team spent a great deal of time researching the Parking Space Detection System. Here is a sample of the study's findings:

Parkopedia: Parkopedia was founded in 2007 with the mission of being able to answer any parking question, anywhere in the world. Today, Parkopedia has become the world's leading parking service provider used by millions of drivers and organizations such as Apple, Audi, BMW, Coyote, Ford, Garmin, GM, HERE, Honda, Jaguar, Land Rover, Mazda, Mercedes-Benz, Peugeot, Porsche, Skoda, Sygic, TomTom, Toyota, Volkswagen, Volvo and others. Parkopedia allows drivers to find the closest parking to their destination, tells them how much it will cost and whether the space is available, which means no more driving around looking for parking! Over the last 10 years, Parkopedia has successfully created a suite of services for its B2B customers and established a winning presence around the world [6].

U-Spot VISIO: Urbiotica's Smart Parking solution revolutionizes urban mobility by significantly cutting down the time spent on the frustrating quest for an available parking space. Innovative technologies, including image processing, are used with the goal of enhancing the overall driving experience. Instantaneous information on parking spaces is provided to drivers by swiftly identifying available spots. By doing so, you simplify the process of looking for parking spots and minimize the time spent on it. Community members are more satisfied and feel better when the positive impact is felt throughout the community. Further, optimizing parking spaces reduces traffic congestion, 5 pollution, and congestion by encouraging more turnover. Citizens, now equipped with the ability to plan their routes more efficiently, contribute to the overall enhancement of city mobility. In addition to supporting park-and-ride facilities, the Smart Parking system enables easy access to local commerce, which is important for local commerce. Urbiotica solution meshes technology with urban planning, enhancing the accessibility of cities, their environmental friendliness, and their residents' well-being [7].

Parktron (IPMS 2000): Founded in 2001, Parktron offers complete solutions and value-added products for intelligent parking management systems. With the advent of new technology and a rise in labor costs, Parktron developed a number of embedded technologies and cloud-based platforms to develop a versatile smart payment kiosk that is suitable for retail, hospitality, hospitals, large campuses, and gas stations, as a result of technological shifts and rising labor costs. Due to the company's integration of advanced technologies, Parktron remains an industry leader in its field, further enhancing its leadership position. As a result of its software solution (PMS2000), the parking facility can be monitored and controlled through a web browser from anywhere in the world. The installation of the system, its configuration, requests for reports regarding revenue generation, and other inquiries can be handled remotely with a few mouse clicks using the system [8].

ii. Requirement Analysis

- **Functional Requirement**

  - The system should provide an admin login interface with fields for username and password entry.

  - The system should detect available parking spaces using video feeds from surveillance cameras.

  - The system should display parking space availability, total number of parked cars, parking lot capacity, and total revenue to the administrators.

  - The system should have logs of Records and display detailed logs of parking space changes, user actions, and system alerts to the administrators.

  - The system should be able to provide a page for entering payment information and processing payments.

- **Non-Functional Requirements**

  - Performance: The system should process and update parking space status in real-time with minimal latency.

  - Usability: The user interface should be simple and easy to navigate for administrators with minimal training.

  - Reliability: The system should be operational 24/7 with minimal downtime for maintenance.

  - Maintainability: The system should be designed for easy updates and maintenance, including the ability to add new features.

  - Security: The system should ensure that only authorized personnel can access sensitive data and functionalities.

b. Feasibility Study

i. Technical Feasibility

The Web application is technically feasible and complies with current technology, including both the hardware and the software. The Web application is supported by almost all the latest web browsers, and most of all, it can run on any modern operating system. Hence, the proposed system is technically feasible. Requirements of our system can be categorized as:

Hardware Requirements:

- A computer with a minimum of 8 GB of RAM and a multi-core processor
- Sufficient RAM for storing and processing image data during the parking space detection.

Software Requirements:

- Operating System: Windows 10, Linux, or MacOS
- Web Browser: Chrome, Firefox, or Safari
- Python 3.6 or above: This is required to run the Flask web framework.
- Flask web framework: This is required to build the web application and API for the system.
- Html/CSS: This is required to build the frontend of the web application and to create a visually appealing user interface.
- JavaScript: JS allows seamless communication between the front-end and back-end components of parking space detection system, which allows us to implement updates, handle user interactions, and provide a more engaging user experience
- Git and GitHub: These are required for version control and collaboration among the development team.

ii. Operational Feasibility

The operational feasibility of the project is moderate, as it involves developing a reliable and accurate parking detection system. The operational feasibility of implementing such a system depends on its seamless integration with existing parking management processes, compatibility with current infrastructure, and user acceptance. Key considerations include regulatory compliance, scalability, reliability, and minimizing operational impact. The system's maintenance requirements, risk management, and customer experience implications also play crucial roles in determining its operational viability. Ultimately, a thorough evaluation of these factors is essential to ascertain the practicality and effectiveness of the parking detection system.

iii. Economic Feasibility

The economic feasibility of the project is high, as the project does not require any significant upfront investment or ongoing operational costs. The required hardware and software resources are typically available on most computers and can be acquired at a relatively low cost. In addition, the development can use open-source software and libraries to reduce the cost of development

iv. Schedule Feasibility

The project expands over a period of 80 days with Tasks being divided along with duration in the time frame. In order for the system to be delivered in the specified time frame we have to make sure deadlines are met and tasks are executed accordingly in the specified duration period.

Table 1: Schedule Planning

S.No	Task Name	Duration	Start Date	End Date
1	Planning	4 Days	14th Aug, 2024	18th Aug, 2024
2	Analysis	6 Days	18th Aug, 2024	23rd Aug, 2024
3	Design	20 Days	23rd Aug, 2024	12th Sep, 2024
4	Coding	25 Days	12th Sep, 2024	7th Oct, 2024
5	Testing	7 Days	7th Oct, 2024	14th Oct, 2024
6	Implementation	8 Days	14th Oct, 2024	22nd Oct, 2024
7	Documentation	87 Days	14th Aug, 2024	9th Nov, 2024

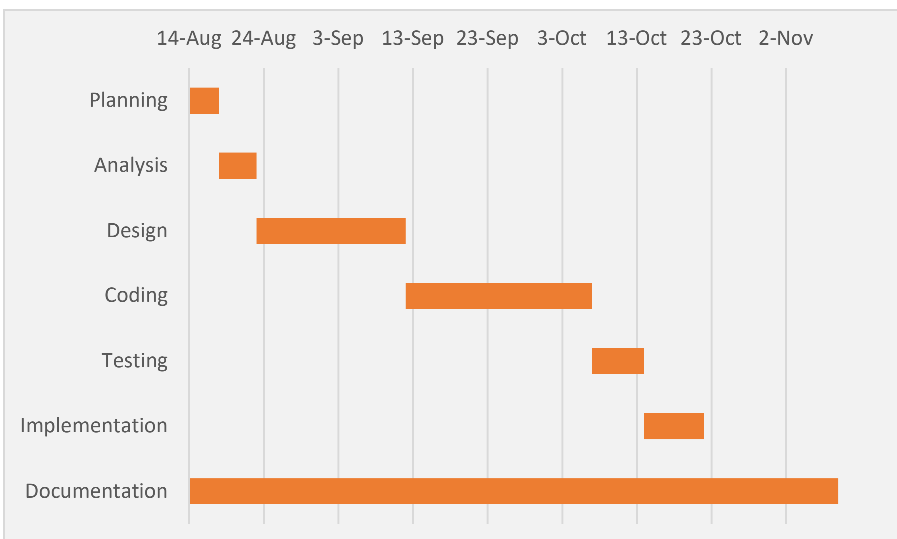

Fig 1. Gantt chart

c. High-Level Design of System

i. Methodology of the proposed system

In the project, the Waterfall method has been used as the development model. This project is a structured and systematic approach, making it suitable when requirements are well-defined, and changes are expected to be minimal. Each phase must be completed before moving on to the next, ensuring a clear progression in the development process.

1. **Requirement Analysis**: In this phase, the requirements for Parking Space Detection are thoroughly defined. This includes specifying the types of vehicles to be detected, the detection zones, and analyzing the features needed to accurately identify and track parked vehicles.
2. **System Design:** Building upon the gathered requirements, the system design phase involves creating a detailed blueprint or architecture for the parking system detection. This includes defining modules, data structures, interfaces, and algorithms crucial for accurately detecting and managing parking spaces.
3. **Implementation (Coding) Phase:** The actual coding or programming of the parking system detection occurs in this phase. Developers use the design specifications to write code for each module or component aimed at effectively detecting and managing parking spots, vehicle entry, and exit.
4. **Testing Phase:** The testing phase ensures that the individual components or modules function correctly when integrated into the complete parking system detection. Various testing methods, such as unit testing and integration testing, are employed to validate the system's functionality and accuracy.
5. **Validation (System Testing) Phase:** The entire parking system detection is thoroughly tested to verify its compliance with the specified requirements. This phase focuses on evaluating the overall functionality and performance to ensure the system reliably detects and manages parking spaces.
6. **Deployment (Implementation) Phase:** Upon successful completion of testing and validation, the Parking Detection System is deployed to the operational environment. This often involves installing the system on designated hardware and making it accessible to end-users.
7. **Maintenance Phase:** The maintenance phase involves providing ongoing support and updates to the Parking Space Detection system. Any identified issues or bugs are addressed promptly, and updates or enhancements may be implemented based on user feedback.

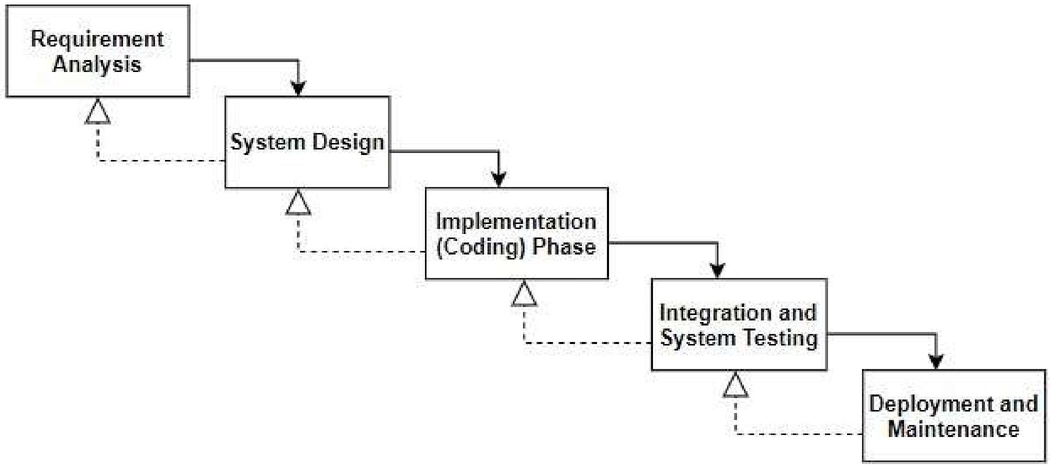

Fig2: Waterfall Model

ii. Flow Charts

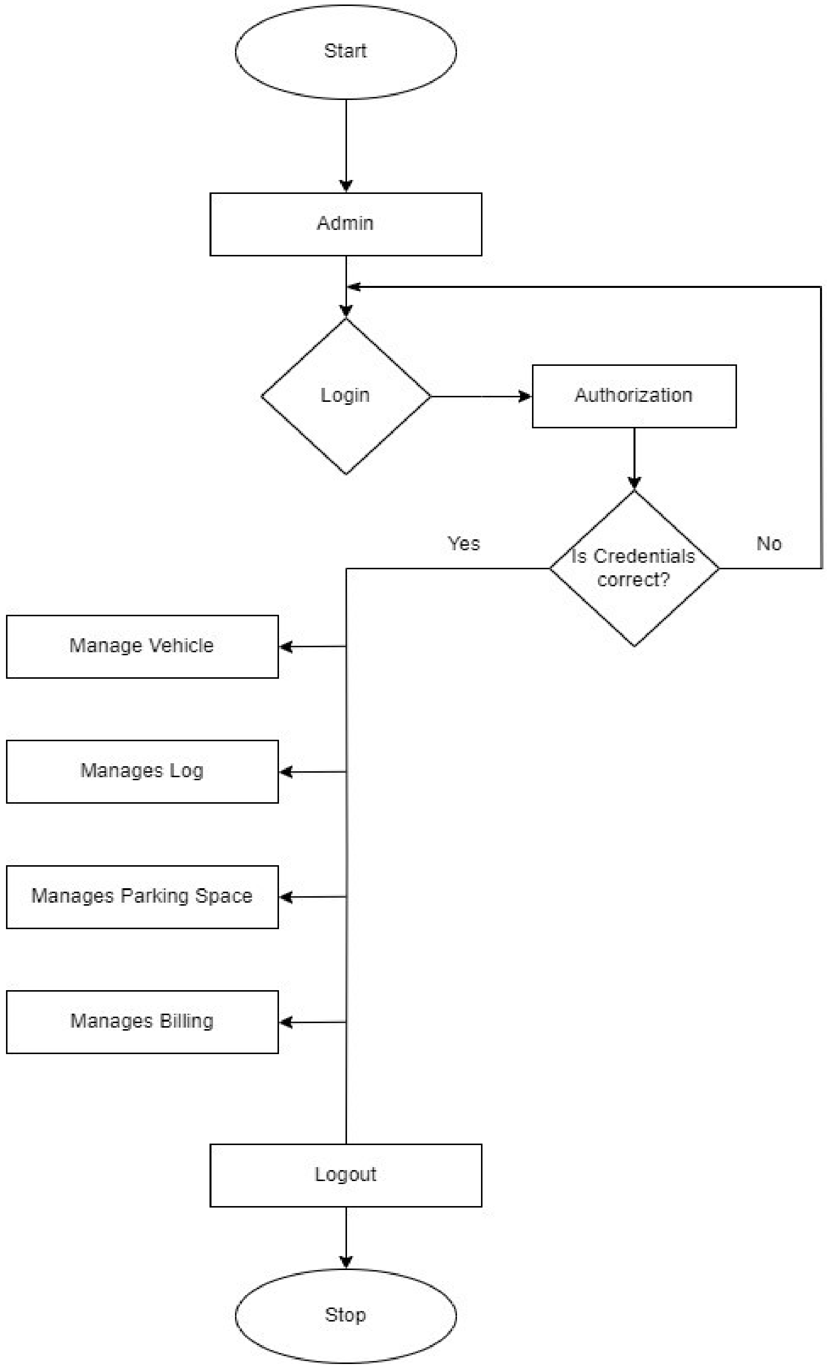

Fig 3: System flow diagram

iii. Working Mechanism of Proposed System

1. **Image Capture:** Cameras strategically placed in the parking lot continuously capture images or video streams of the entire area.

2. **Vehicle Detection**: OpenCV processes these images using techniques like contour detection, and edge detection, to recognize vehicles and distinguish between occupied and vacant parking spaces.

3. **Data Analysis**: Python scripts analyze the processed data, keeping an accurate count of available and occupied parking spaces. This data is stored and updated in real time to reflect the current parking status.

4. **Web Interface**: Flask serves the analyzed data to users through a responsive web application, which can be accessed via browsers or mobile devices. The interface displays the total number of parking spaces, available spots, and their locations, guiding users to the nearest free space.

iv. Description of Algorithm

- Step 1: Start the procedure.
- Step 2: Capture images or video data manually or from datasets using the OpenCV library.
- Step 3: Find x and y coordinates and determine the coordinates of parking spaces.
- Step 4: Initialize the space counter by setting the space counter to zero.
- Step 5: Compare obtained and assumed values. If they match, end the procedure. Else, proceed to frame selection.
- Step 6: select the coordinates (width and height).
- Step 7: Detect edges of parking slots.
- Step 8: Compare edge detection results. If edge detection matches the assumed value, proceed. Else, compare obtained and assumed values.
- Step 9: Detect occupied slots and use a threshold value to determine if slots are occupied.
- Step 10: Mark free slots. If occupied slots are fewer than the threshold, mark them as free.
- Step 11: compare values again and compare obtained and assumed values again.
- Step 12: Mark occupied slots and if occupied slots exceed the threshold, mark the slot as occupied. Else, mark it as free.
- Step 13: End the procedure.

Algorithm used for Edge Detection (Canny Edge Detection Algorithm)

Step 1: Gaussian Smoothing: The first step involves smoothing the image using a Gaussian filter to reduce noise and suppress unwanted details.

Step 2: Gradient Calculation: Compute the image gradients in both the horizontal (*Gx*) and vertical (*Gy*) directions using Sobel or other gradient operators. Also, calculate the gradient magnitude *M* and the gradient direction θ for each pixel. While using Sobel Filter, you can select kernel matrix size 3 or 5.

$$M = \sqrt{(G_x)^2 + (G_y)^2}$$
$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

Step: Non-Maximum Suppression: Suppress non-maximum values in the gradient magnitude image. Only local maxima along the gradient direction are retained.

Step 4: Thresholding: The dual-threshold hysteresis approach helps to reduce the impact of noise and produce a more robust edge map.

Step 5: Edge Tracking by Hysteresis: Utilize two thresholds, a high threshold (T-high) and a low threshold (T-low), to classify pixels as strong, weak, or non-edges.

5. **Expected Outcomes**

1. **Efficient Parking Management**:
    a. Streamlined parking operations with real-time monitoring of parking space availability.
    b. Enhanced ability for parking lot managers to optimize space usage, reducing underutilization or overcrowding.
2. **Reduced Traffic Congestion**:
    a. Minimized traffic jams caused by vehicles searching for parking, leading to smoother traffic flow in and around parking facilities.
    b. Lowered environmental impact due to reduced vehicle idling and circling for parking.
3. **Improved User Experience**:
    a. Drivers experience less frustration and stress, finding parking spots quickly and easily with real-time guidance.
    b. Increased customer satisfaction for businesses and venues, as convenient parking encourages repeat visits.
4. **Data-Driven Insights**:
    a. Collection of valuable data on parking patterns and usage trends, helping managers make informed decisions about pricing, space allocation, and future expansion.
    b. Ability to forecast demand and implement dynamic pricing or reservation systems based on real-time and historical data.
5. **Scalability and Adaptability**:
    a. A system that can be easily scaled to accommodate larger or multiple parking facilities, with the flexibility to adapt to different types of parking environments (e.g., outdoor lots, underground garages).
    b. The potential for integration with other smart city solutions, such as traffic management systems, to further enhance urban mobility.
6. **Cost Savings**:
    a. Reduction in operational costs due to better space management and decreased need for manual monitoring.
    b. Potential revenue growth through optimized parking space utilization and enhanced customer satisfaction.

6. References

[1] Carnelli, Pietro, et al. "Parkus: A novel vehicle parking detection system." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 31. No. 2. 2017 [Accessed August 16, 2024].

[2] Q. Wu, C. Huang, S. -y. Wang, W. -c. Chiu and T. Chen, "Robust Parking Space Detection Considering Inter-Space Correlation," 2007 IEEE International Conference on Multimedia and Expo, Beijing, China, 2007, pp. 659-662, doi: 10.1109/ICME.2007.4284736 [Accessed August 16, 2024].

[3] M. Y. I. Idris, E. M. Tamil, N. M. Noor and K. W. Fong, “Parking Guidance System Utilizing Wireless Sensor Network and Ultrasonic Sensor”, Information Technology Journal, ISSN 1812-5638, (2009) [Accessed August 17, 2024].

[4] Abrar Fahim, Mehedi Hasan, Muhtasim Alam Chowdhury,Smart parking systems: comprehensive review based on various aspects,Heliyon,Volume 7, Issue 5,2021,e07050,ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2021.e07050 [Accessed August 17, 2024].

[5] A. Kianpisheh, N. Mustaffa, J. M. Y. See and P. Keikhosrokiani, “User Behavioral Intention toward Using Smart Parking System”, Proceeding of ICIEIS, (2011) Kuala Lumpur, Malaysia, pp. 732-747 [Accessed August 18, 2024].

[6] Eugene Tsyrklevich, Parkopedia https://www.parkopedia.com/ [online] [Accessed August 18, 2024].

[7] Ubiotica, U-Spot VISIO https://urbiotica.com/en/hardware-product/u-spotvisio-a-i-single-space-detection/ [online] [Accessed August 19, 2024].

[8] Parktron, (IPMS 2000)http://www.parktron.com/enus/products/detail/263?page=1 [online] [Accessed August 21, 2024].