PROJECT REPORT

1. INTRODUCTION

1.1 Project Overview

The AI-enabled car parking system utilizing OpenCV combines the power of computer vision and artificial intelligence to automate and optimize the process of parking cars. By leveraging OpenCV's image processing and object detection capabilities, this project aims to provide an intelligent solution for efficient parking space management.

Traditionally, parking lots often face challenges such as limited parking spaces, congestion, and inefficient vehicle allocation. This project addresses these issues by automating the parking process, allowing for better space utilization and reduced human intervention.

The core technology employed in this project is OpenCV, an open-source computer vision library that provides a rich set of functions and algorithms for image and video analysis. OpenCV allows us to process real-time video streams from parking lot cameras and perform object detection to identify cars.

In conclusion, the AI-enabled car parking system using OpenCV aims to revolutionize the traditional parking experience by leveraging computer vision and AI technologies. By automating parking space management, it improves efficiency, optimizes resource utilization, and enhances the overall parking experience for both drivers and parking lot operators.

1.2 Purpose

The main objectives of the AI-enabled car parking system using OpenCV are:

1. Vehicle Detection: Utilizing image processing techniques and machine learning algorithms, the system detects and recognizes cars in real-time video streams. This allows for the identification and tracking of available parking spaces.

2. Parking Space Allocation: The system analyzes the occupancy status of parking spaces based on the detected vehicles. It determines which parking spaces are available and which are occupied, providing real-time information to drivers searching for parking spots.

3. Parking Guidance: By leveraging computer vision and AI, the system can guide drivers to available parking spaces using visual indicators or digital displays. This helps to reduce the time spent searching for parking and minimizes congestion within the parking lot.

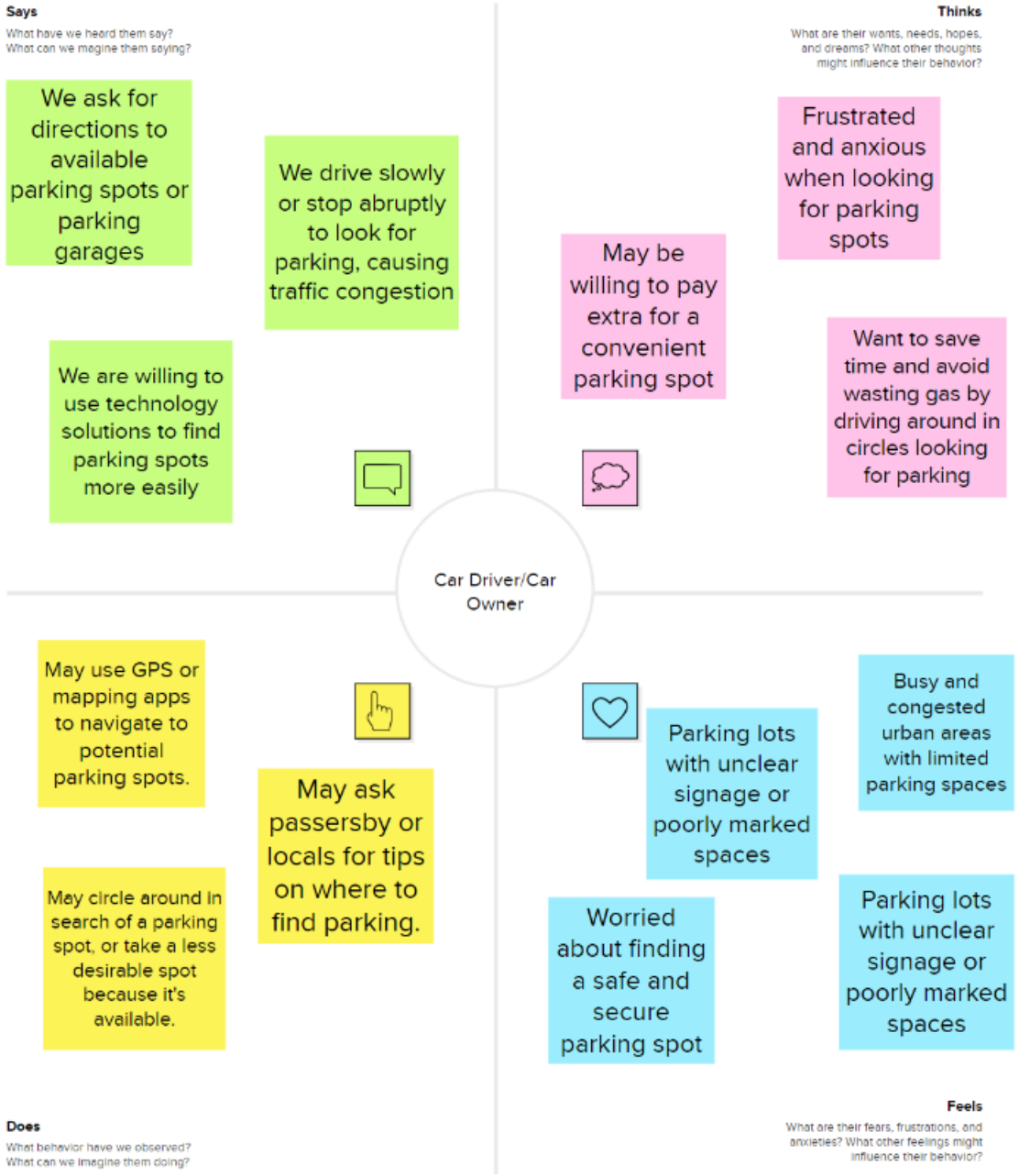
4. Security and Monitoring: The AI-enabled car parking system can incorporate additional features such as license plate recognition, surveillance, and monitoring of parking areas. This enhances security and allows for effective enforcement of parking regulations.

2. IDEATION & PROPOSED SOLUTION

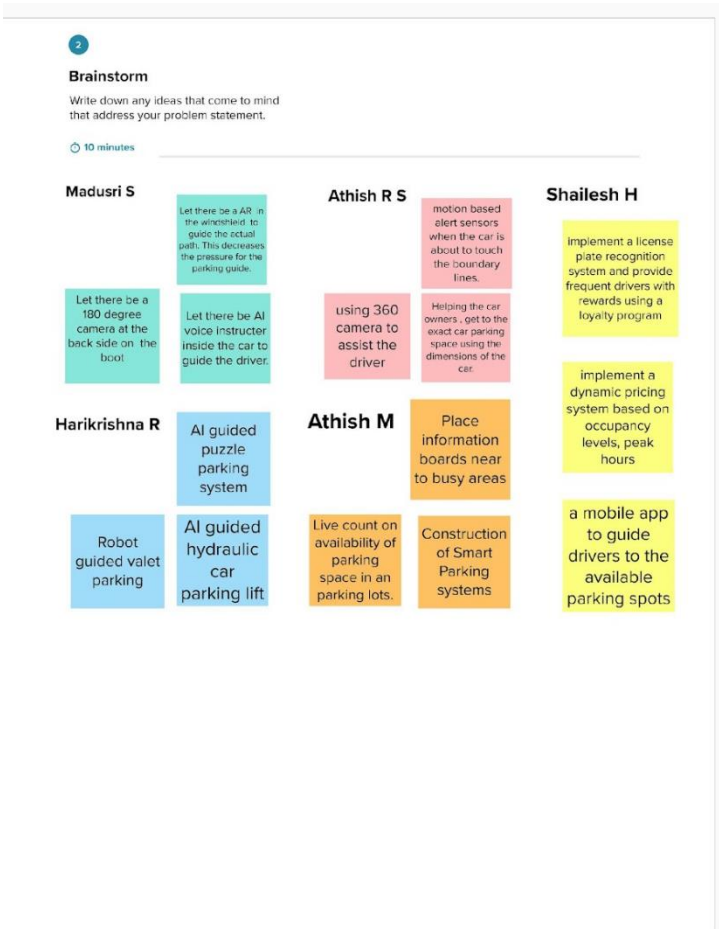
2.1 Problem Statement Definition

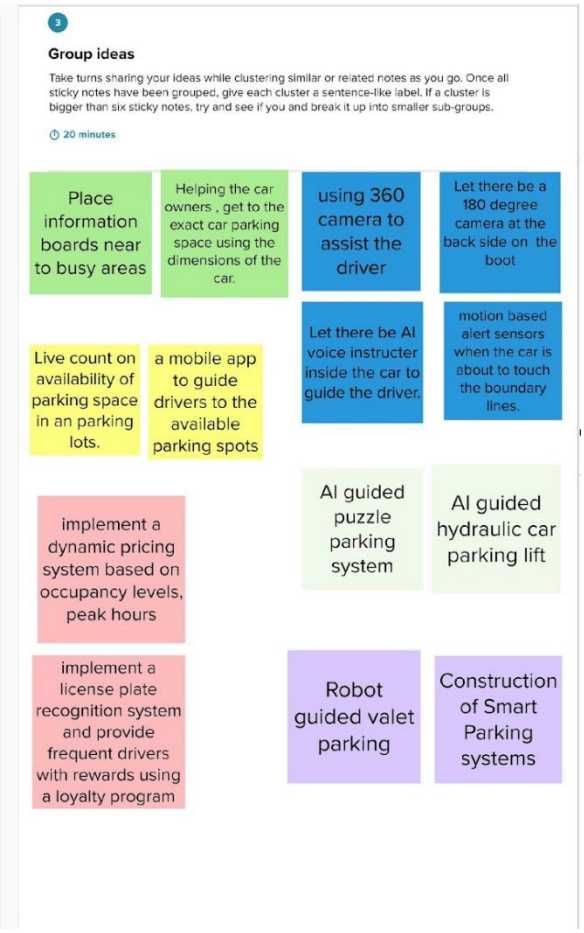
The aim of this project is to develop a real-time AI enabled car park assit for drivers that accurately determines the no of available parking spaces and the number of free parking spaces respectively. This Project will be able to demostrate graphically the parking space to the user. So,that it becomes convinent of the driver to park the car ,without wasting time and other resources.The system must function effectively in various lighting conditions and be capable of operating under different weather conditions.

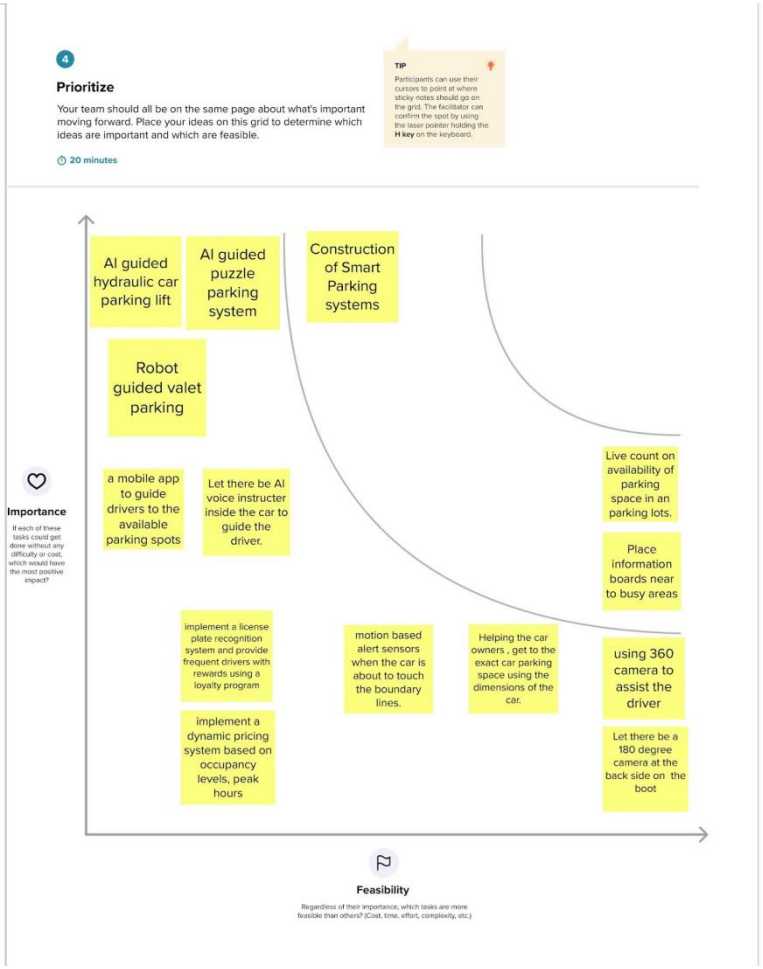
2.2 Empathy Map Canvas



2.3 Ideation & Brainstorming







2.4 Proposed Solution

S.No.	Parameter	Description
	Problem Statement (Problem to be solved)	Finding a parking space in metro cities is an uphill task. Some people thus prefer to take cabs, instead of their own vehicles, to avoid the hassle of searching for a parking spot and the possibly running late. Also, parking at street corners leaves vehicles unguarded from potential damage by miscreants or natural causes.
	Idea / Solution description	To design and develop a system that uses computer vision techniques to automatically detect and allocate empty parking spots in real-time. The system will use cameras to capture images of the parking area and process them using OpenCV to identify available parking spots. The system will then direct drivers to empty spots through an intuitive user interface, reducing the time and effort required for parking.
	Novelty / Uniqueness	This system is different from traditional parking systems as it reduces the need for manual labor and provides an intuitive user interface that can help drivers find available parking spots easily. Furthermore, this system can be integrated with other smart city technologies, such as traffic management systems and smart lighting systems, to provide a comprehensive solution for managing parking in urban areas. The use of computer vision and AI technologies to automate the parking process is a unique approach that can improve the overall parking experience for drivers while reducing the time and effort required for parking.
	Social Impact / Customer Satisfaction	The project has a significant social impact by providing an efficient and automated parking solution. It can help reduce the time and effort required for parking, which can result in reduced traffic congestion and lower emissions. The system can also improve the overall parking experience for drivers, resulting in increased customer satisfaction. The system's intuitive user interface can help drivers find available parking spots easily, which can reduce frustration and stress associated with finding parking in crowded urban areas. Moreover, the automated system can help reduce the number of vehicles circling around in search of parking, leading to lower traffic congestion and emissions.
	Business Model (Revenue Model)	The project can be developed into a viable business model by offering it as a service to parking lot owners and operators. The system can be deployed in parking lots, garages, and other parking facilities, where it can automate the parking process and provide an efficient parking solution. The business model could be based on a subscription-based service or a pay-per-use model, where parking lot owners pay a fee based on the number of parking spaces the system manages or the number of parking sessions the system handles. The subscription-based model could have different tiers, depending on the number of parking spaces the parking facility has or the number of features and services provided by the system.
	Scalability of the Solution	The scalability of the AI-enabled car parking system using OpenCV is high. The system can be deployed in parking lots of varying sizes, from small lots with just a few parking spaces to large multi-story garages with hundreds or even thousands of parking spaces. The system's scalability is due to the fact that it relies on computer vision and AI algorithms, which are highly adaptable and can be easily customized to fit different parking lot configurations and sizes. Additionally, the system can be integrated with other smart city technologies, making it even more scalable and adaptable to changing urban environments.

3. REQUIREMENT ANALYSIS

3.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through the website
FR-2	User Confirmation	Confirmation via Email, Phone number Confirmation via OTP
FR-3	Highlighted Parking Availability Section	to know the total number of parking spaces in the parking area and the available number of spaces.
FR-4	Availability Status Notification	Facilitating User Awareness of Availability
FR-5	Radio buttoned GUI	Enabling the user to secure a singular position within the available space
FR-6	Confirmation popup	Ensuring the user's allocation.

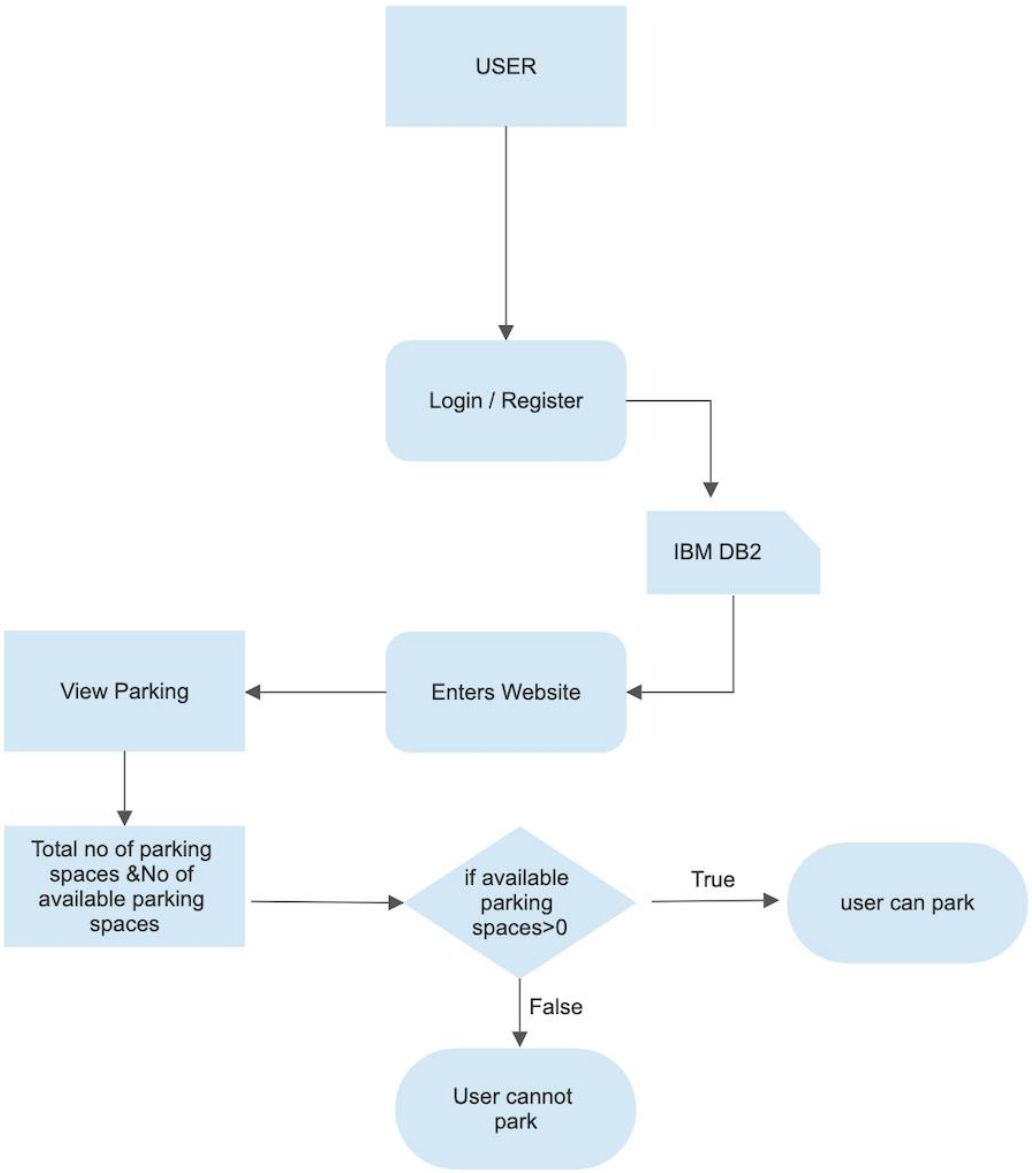
3.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

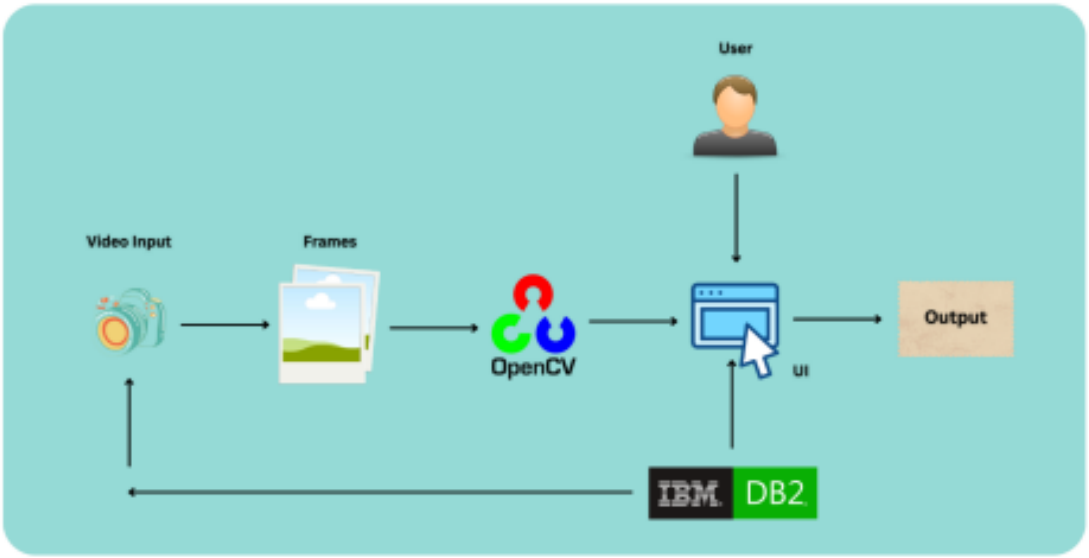
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Intuitive and Accessible User Interface (UI)
NFR-2	Security	Secured Login Credentials with Password and Registration Confirmation OTP
NFR-3	Reliability	Real-time Database Management System Updates
NFR-4	Performance	LTE-powered website, Seamless, high-speed connectivity for enhanced digital experience.
NFR-5	Availability	An optimized asset-driven website: Fast, responsive, and visually captivating
NFR-6	Scalability	Easily scales to accommodate growing user demand.

4. PROJECT DESIGN

4.1 Data Flow Diagram



4.2 Solution Architecture



4.3 User Stories

- As a user, I can register in the website by providing credentials such as email, mobile number.
- As a user, I want to access the Parking area's structure.
- As a user, I want to know the total no of parking spaces in the parking area and also the available no of spaces.
- The task is to highlight the free parking spaces in the parking space inorder for the user to choose their parking position.
- As a user, I **am** obligated to know whether i will be able to park my car in the parking or not.If not I must be notified.

5. CODING & SOLUTIONING

Flask Web Application: The code sets up a Flask web application using the Flask class from the Flask framework. It handles different routes and serves HTML templates to the client.

Route Definitions: The code defines several routes using the @app.route() decorator. Each route corresponds to a specific URL path and specifies which HTML template to render when that route is accessed.

HTML Template Rendering: The render_template() function is used to render HTML templates. The 'index.html' and 'model.html' templates are rendered when the respective routes are accessed.

Video Processing: The code utilizes OpenCV (cv2) to process a video feed. It reads frames from a video file named 'carParkingInput.mp4' using cv2.VideoCapture(). The frames are then processed to detect parking spaces.

Parking Space Detection: The code loads a list of parking space positions from a file named 'parkingSlotPosition' using the pickle module. For each position, it extracts the corresponding region of interest (ROI) from the processed frame and performs calculations to determine if the space is occupied or vacant. The occupancy status is visualized by drawing rectangles on the frame using cv2.rectangle().

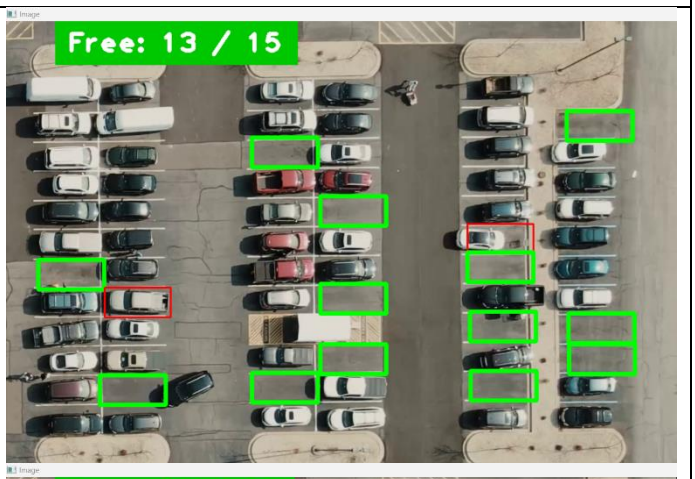
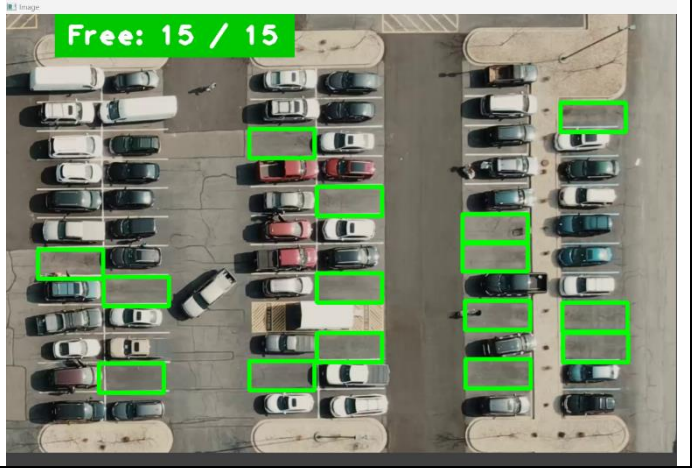
Displaying the Processed Frame: The processed frame, with parking space rectangles and occupancy information, is displayed using cv2.imshow(). Various image processing techniques such as grayscale conversion, blurring, thresholding, and dilation are applied to the frame before parking space detection.

Infinite Loop: The processing and display of frames continue in an infinite loop until the user presses the 'q' key to exit the loop.

Flask App Execution: The Flask application is run using app.run(debug=True), which starts the development server and makes the application accessible.

6. RESULTS

6.1 Performance Metrics

S.No.	Parameter	Values	Screenshot
1.	Accuracy		 

7. ADVANTAGES & DISADVANTAGES

Advantages:

Accuracy: OpenCV car parking systems can provide accurate detection and tracking of vehicles within a parking lot. The computer vision algorithms used in OpenCV can efficiently identify and analyze vehicle shapes, sizes, and movements, leading to accurate parking space occupancy detection.

Real-time monitoring: OpenCV car parking systems can operate in real-time, allowing parking lot operators to monitor the occupancy status and availability of parking spaces in real-time. This information can be conveyed to drivers through digital signage or mobile applications, enabling them to find available parking spaces quickly.

Cost-effective: OpenCV is an open-source library, making it cost-effective for implementing car parking systems. It eliminates the need for expensive proprietary software and licenses, reducing the overall cost of the system.

Flexibility: OpenCV provides a wide range of image processing and computer vision functions, allowing developers to customize and adapt the car parking system according to specific requirements. It offers flexibility in terms of algorithm selection, parameter tuning, and integration with other software components.

Disadvantages:

Complex setup and configuration: Setting up an OpenCV car parking system can be challenging and requires expertise in computer vision and image processing. Configuring the system to work accurately in different lighting conditions, camera angles, and parking lot layouts may involve trial and error.

Sensitivity to environmental factors: OpenCV car parking systems heavily rely on camera inputs for vehicle detection and tracking. Factors such as lighting variations, shadows, occlusions, and adverse weather conditions (rain, snow) can affect the system's performance and accuracy. Extra measures may be required to mitigate these challenges.

Hardware requirements: Running an OpenCV car parking system typically requires a computer or embedded device with sufficient processing power and memory. Depending on the complexity of the algorithms and the number of cameras used, the hardware requirements can be significant, leading to additional costs.

Maintenance and updates: OpenCV car parking systems require regular maintenance and updates to ensure optimal performance. Camera calibration, software updates, and periodic checks are necessary to address any issues and keep the system functioning accurately.

8. CONCLUSION

The project report titled "AI-Enabled Car Parking Using OpenCV" presents a comprehensive study and implementation of an innovative solution for efficient car parking management. By leveraging AI and computer vision techniques through OpenCV, the project aimed to address challenges in traditional car parking systems. The implementation involved cameras, image processing algorithms, and real-time information for drivers. Achievements include vehicle detection, parking space analysis, enhanced security, and scalability potential. Overall, the project demonstrates a successful intelligent car parking system.

9.FUTURE SCOPE

The future scope of OpenCV car parking systems is promising, with potential advancements and applications in several areas. Here are some future possibilities:

Advanced object detection and recognition: OpenCV car parking systems can benefit from advancements in object detection and recognition algorithms. Future developments may include more accurate and efficient algorithms for detecting vehicles, as well as the ability to recognize and differentiate between different types of vehicles, such as cars, motorcycles, buses, and trucks.

Integration with smart parking management systems: OpenCV car parking systems can be integrated with smart parking management systems to provide a comprehensive parking solution. This integration can enable features such as real-time parking space availability updates, automatic parking fee calculation, mobile app integration for parking reservations, and guidance to the nearest available parking space.

Deep learning and neural networks: The integration of deep learning techniques and neural networks can enhance the capabilities of OpenCV car parking systems. These technologies can improve the accuracy of vehicle detection and tracking, handle complex scenarios and occlusions, and provide robust performance in various environmental conditions.

Integration with IoT and cloud computing: OpenCV car parking systems can be integrated with IoT (Internet of Things) devices and cloud computing platforms to enable efficient data collection, storage, and analysis. This integration can facilitate real-time monitoring, data analytics, and the implementation of predictive parking analytics for better parking management and optimization.

Automated payment and ticketing systems: Future OpenCV car parking systems may incorporate automated payment and ticketing systems. This can include features such as license plate recognition for automatic payment, digital ticketing through mobile apps, and seamless integration with payment gateways or electronic wallets for convenient and contactless transactions.

Enhanced security and surveillance: OpenCV car parking systems can play a significant role in enhancing security and surveillance in parking lots. Future advancements may include features like suspicious activity detection, vehicle anomaly detection, and integration with security systems for proactive response to security threats.

Integration with smart city infrastructure: OpenCV car parking systems can be integrated into smart city infrastructure to contribute to overall urban management. This integration can involve data sharing with traffic management systems, intelligent routing algorithms, and integration with smart transportation systems for efficient traffic flow and reduced congestion.

10.APPENDIX

Source Code : **Sourcecode_link**

GitHub & Project Video Demo Link

Github

demo_video