

## **AI Enable car parking using OpenCV**

Submitted by

Sai thanmay.A (21BCE7877)

Manishwar.K (21BCE7185)

Sai Gopal.P (21BCE8548)

Varshith.C (21BCE7733)

## **INTRODUCTION**

### **1.1 Project Overview**

Most AI-enabled car parking systems parking a car today are still managed by hand. There is no automated monitoring system in place to keep track of how much capacity each parking place contains. In order to find an empty spot, drivers often have to make a circuitous trip through the parking lot. Where there are more people than parking spots, such problems are especially common near hospitals, malls, schools, and other large gathering places.

### **1.2 Purpose**

The purpose of using AI-enabled car parking systems with OpenCV (Open Source Computer Vision) is to automate and optimize the process of parking vehicles in a parking lot or garage. By employing computer vision techniques and AI algorithms, these systems can efficiently detect and track vehicles, analyze parking space availability, and guide drivers to vacant spots.

## 2.1 Problem Statement Definition

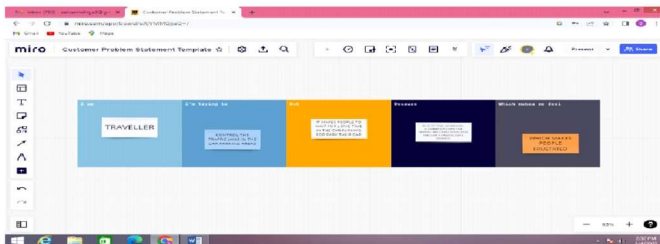
### Customer Problem Statement template:

Problem statement 1 :

Consequently, once a car enters a parking garage followed by a parking space, a ping ultrasonic sensor will then be able to determine if a car is parked in the space or not. This information would then be relayed to update the network.

Problem statement 2 :

Avoid excessive parking supply. Use Parking Management to encourage more efficient use of existing parking facilities and address any spill over problems that result from pricing. Develop Transportation Management Associations to provide parking management and brokerage services in a particular area.



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Traveller	Control the traffic in car parking	It makes more time	Due to the increased number of cars	Frustrated
PS-2	Traveller	Make the traveller to feel automated	It makes more time	Increased traffic	Frustrated

## 2.2 Empathy Map Canvas

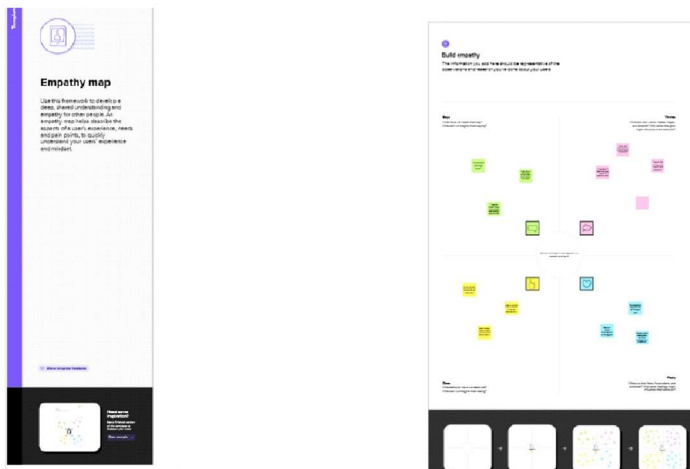
## Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

### EXAMPLE:



## 2.3 Ideation & Brainstorming

### Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference: <https://www.mural.co/templates/empathy-map-canvas>

#### Step-1: Team Gathering, Collaboration and Select the Problem Statement

**Brainstorm & Idea Prioritization**

Use this template in your room or online to brainstorm ideas and prioritize them. Get everyone's input and ideas, and use them to create a rich amount of creative solutions.

1. Brainstorm ideas  
2. Prioritize ideas  
3. Collaborate on ideas

**Before you collaborate**

1. Define the problem  
2. Brainstorm  
3. Prioritize  
4. Collaborate

**Define your problem statement**

1. Define the problem statement  
2. Define the problem statement

**My idea of brainstorming**

1. Brainstorm ideas  
2. Prioritize ideas  
3. Collaborate on ideas

## Step-2: Brainstorm, Idea Listing and Grouping

2

### Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

**TIP**  
You can select a sticky note and hit the pencil icon (switch to sketching) icon to start drawing!

Person 1

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 2

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 3

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 4

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 5

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 6

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 7

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

Person 8

1. I think we should use a camera to detect the car's position and then use a sensor to detect the car's speed. This way we can know exactly where the car is and how fast it is moving.

## Step-3: Idea Prioritization

3

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

#### TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

**Image processing** The images returned by the camera will be processed using OpenCV to identify the location of parked vehicles and available parking spaces. The next step is to identify the location of the vehicle in the image.

**Object detection** Using object detection techniques such as Haar cascades or YOLO, the system can identify the location of vehicles in the parking lot.

**Classification** The system can classify the identified objects as cars or other objects to ensure that only cars are considered as possible parking spaces.

**Availability detection** After identifying the vehicles, the system can analyze the parking spaces to determine which ones are available and which ones are occupied.

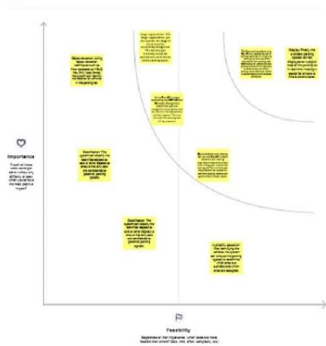
**Display** Finally, the available parking spaces can be displayed on a digital map of the parking lot in real-time, making it easier for drivers to find a parking spot.

The AI-enabled parking system can be used to identify the location of parked vehicles and available parking spaces in real-time, making it easier for drivers to find a parking spot.

### Prioritize

You represent each idea on the same page with sticky notes. Now, you can group and prioritize ideas by moving them around the page. Use the following steps to prioritize ideas in your mural and create a final list.

10 minutes



### After you collaborate

You can explore how we use ideas in our system. You can also explore our system's features and capabilities.

#### Quick actions

- 1. **Search for ideas** Find ideas related to your current project or topic.
- 2. **Filter ideas** Filter ideas by status, date, or other criteria.
- 3. **Export ideas** Export ideas to a file or database.

#### Steps to create an idea

1. **Define the problem** Clearly define the problem you are trying to solve.
2. **Generate ideas** Brainstorm ideas and solutions.
3. **Evaluate ideas** Evaluate ideas based on feasibility and importance.
4. **Implement ideas** Implement the selected ideas.

#### More helpful features

## 2.4 Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No. Parameter Description

### 1. Problem Statement (Problem to be solved)

Traditional car parking systems are inefficient and time-consuming. Drivers often have to spend a lot of time searching for an empty parking space, leading to congestion in the parking lot. Additionally, traditional parking systems require manual intervention for ticketing and payment processing, which can lead to long queues and delays.

2. Idea / Solution description The idea behind an AI-enabled car parking system using OpenCV is to leverage the power of computer vision to automate the parking process. By using cameras and image processing algorithms, the system can detect the presence of cars in the parking lot and guide drivers to empty parking spaces. The system can also automate the ticketing and payment processing, making the process faster and more convenient for drivers.

3. Novelty / Uniqueness The use of computer vision technology for car parking systems is a novel approach that can significantly improve the efficiency and convenience of parking. By using OpenCV, the system can accurately detect and track cars, even in crowded parking lots. Additionally, the use of AI algorithms can optimize the parking process by analyzing parking patterns and occupancy rates

4. Social Impact / Customer Satisfaction An AI-enabled car parking system using OpenCV can have a significant social impact by reducing traffic congestion and improving the overall driving experience. By guiding drivers to empty parking spaces, the system can reduce the time spent searching for parking and minimize traffic congestion in the parking lot. Additionally, by automating the ticketing and payment process, the system can improve the convenience and efficiency of the parking process.



5. **Business Model (Revenue Model)** The business model for an AI-enabled car parking system can be based on a pay-per-use model, where drivers pay a fee based on the duration of the parking. The system can be integrated with a payment gateway to enable online payments, making the process faster and more convenient for drivers. Additionally, the system can generate revenue by collecting data on parking patterns and occupancy rates, which can be used by parking lot operators to optimize the parking process.

6. **Scalability of the Solution** An AI-enabled car parking system using OpenCV can be easily scaled to accommodate large parking lots and multiple parking locations. By using a centralized database and cloud-based architecture, the system can easily handle large volumes of data and support multiple users. Additionally, the system can be customized based on the specific requirements of the parking lot, making it adaptable to different environments and use cases.

### 3. REQUIREMENT ANALYSIS

#### 3.1 Functional requirement

##### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User authentication	Capture and store user's face image  Authenticate user's identity based on face recognition  Allow access to parking lot upon successful authentication

FR-2	Parking spot management	<p>Detect availability of parking spots using camera feed</p> <p>Assign parking spot to the user</p> <p>Update parking spot availability in real-time</p>
FR-3	Payment and billing	<p>Calculate parking fee based on parking duration</p> <p>Allow payment via various modes like credit card, mobile wallet</p>
FR-4	Security and surveillance	<p>Monitor parking lot using cameras for suspicious activities</p> <p>Alert security personnel in case of any security breach</p>
FR-5	System maintenance and support	<p>Provide regular maintenance and updates to the system</p> <p>Offer customer support for any issues faced by users</p>

## 3.2 Non-Functional requirements

### Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should be easy to use and user-friendly, providing clear instructions to drivers on where to park and displaying available spots in an intuitive manner.
NFR-2	Security	The system should be secure and able to prevent unauthorized access to the parking lot. It should also protect the data collected by the system, ensuring that it cannot be accessed by unauthorized parties.

NFR-3	<b>Reliability</b>	The system must be reliable and accurate in detecting and identifying available parking spots. It should also be able to identify and prevent unauthorized access to the parking lot.
NFR-4	<b>Performance</b>	The system must be fast enough to detect and identify available parking spots in real-time, allowing drivers to quickly find a parking spot. It should also be able to handle multiple vehicles entering and exiting the parking lot simultaneously.
NFR-5	<b>Availability</b>	The system should be compatible with different types of vehicles, including cars, trucks, and motorcycles, and be able to accommodate different sizes of vehicles.
NFR-6	<b>Scalability</b>	The system should be able to handle a large number of parking spaces, and be easily scalable to accommodate additional spaces in the future.

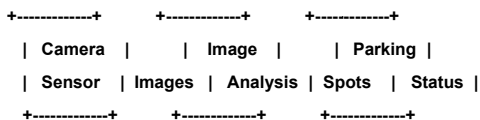
## 4. PROJECT DESIGN

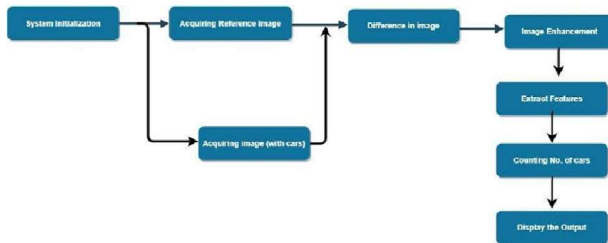
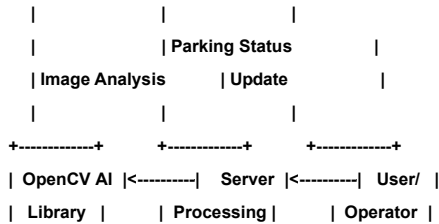
### 4.1 Data Flow Diagrams

#### Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: [\(Simplified\)](#)





## 4.2 Solution & Technical Architecture

### Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



Example - Solution Architecture Diagram:

**Figure 1: Architecture and data flow of the car parking using open cv**

#### 4.3 User Stories

**User Stories**

Use the below template to list all the user stories for the product.

Customer (driver)	Registration	USN-1	As a driver, I want to be able to enter the parking lot and have the system detect a parking space so that I can park my car.	The algorithm can detect empty parking spaces with an accuracy rate of at least 95%.	High	Ragul
		USN-2	As a driver, I want to be able to reserve a parking space so that I am guaranteed a spot when I arrive.	The reservation system can handle multiple reservations at the same time.	Medium	Ragul
		USN-3	As a parking lot attendant, I want to be able to monitor the parking lot in real-time so that I can manage the parking spaces effectively.	The system provides real-time updates of occupied and available parking spaces to the attendant.	High	Ragul
		USN-4	As a parking lot owner, I want to be able to generate reports on parking lot usage so that I can make informed decisions about pricing and capacity.	The reporting system can generate reports on parking lot usage by hour, day, and month.	Medium	Ragul
	Traveller	USN-5	As a parking lot user, I want to be able to pay for my parking spot using my mobile device so that I don't have to use cash or a credit card.	The mobile payment system is secure and uses encryption to protect user information.	Low	Sampath
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						