Minor Project

SMART PARKING MANAGEMENT SYSTEM

Presented By:
Shreeya Jaiswal
Sarthak Jain
Vansh Grover

Mentored By:
Mr. Amit Panwar

Problem Statement

- Inefficiency of Current Parking Systems: Drivers spend a significant amount of time searching for parking in crowded urban areas, leading to unnecessary congestion, delays, and increased frustration among commuters.
- Lack of Visibility in Parking Facilities: Many public parking lots and malls lack systems to indicate free and occupied spaces, forcing drivers to circle around, which contributes to traffic buildup and longer waiting times.
- Environmental Impact: Prolonged vehicle idling while searching for parking increases fuel consumption and carbon emissions, leading to a negative impact on the environment and public health.
- Outdated Technology: Traditional parking systems in many cities do not use smart technologies like real-time monitoring or automation, resulting in inefficient parking management and missed opportunities for improvement.
- Proposed Smart Parking Solution: Our system employs image processing to detect vehicle occupancy and uses green and red lights to indicate availability. By making the system open-source, we aim to promote affordability, accessibility, and community collaboration for future enhancements.

Objective

The primary objective of this project is to create an open-source Smart Parking System that leverages image processing and real-time signaling technology to improve parking efficiency in urban environments. This system will accurately detect vehicle occupancy in parking spaces, reducing the time drivers spend searching for available spots —green for free spots and red for occupied ones. By minimizing search time, the system will help reduce traffic congestion, improve overall traffic flow, and lower emissions from idle vehicles

Area of Application (AOA)

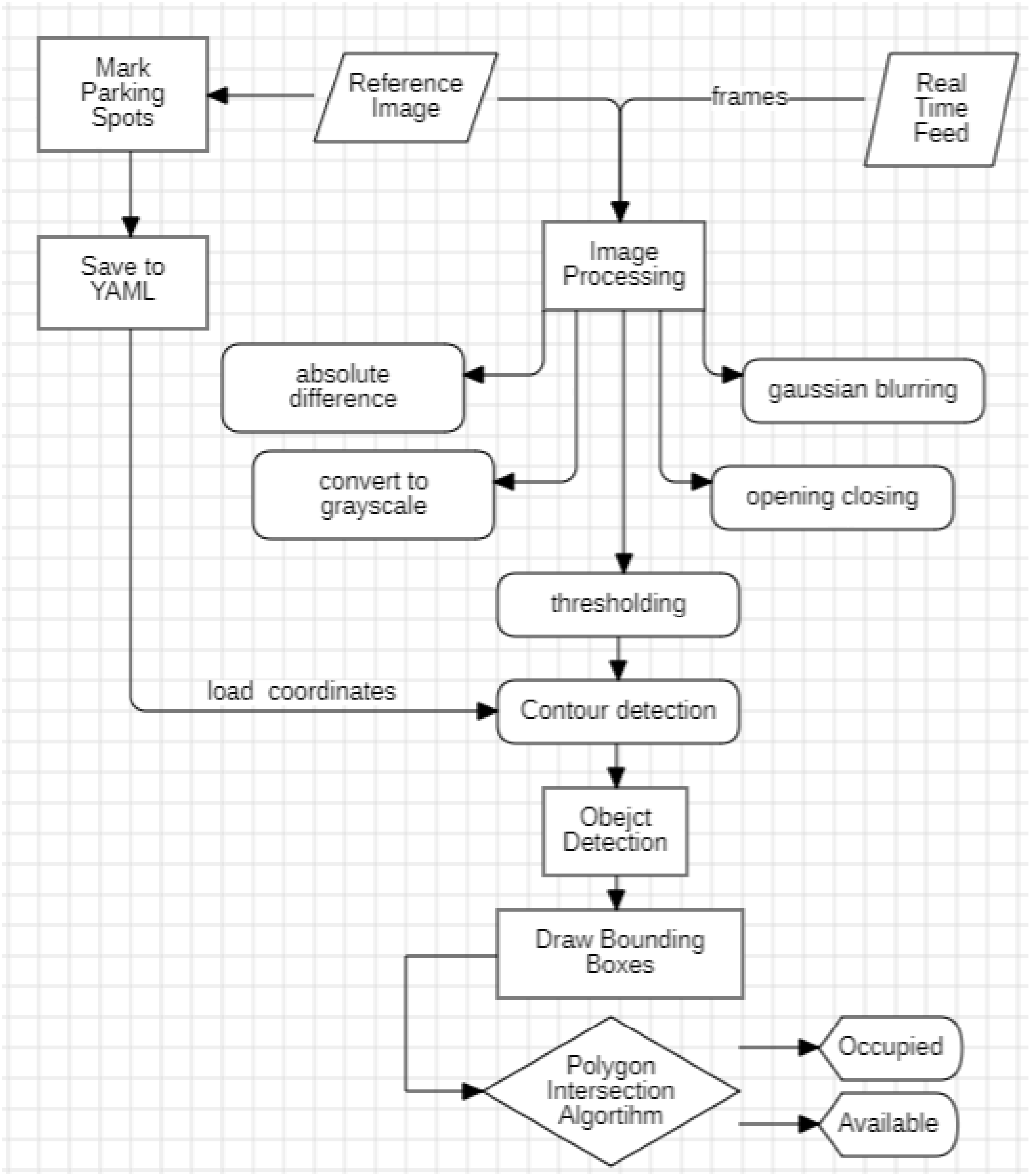
1. Shopping Malls and Retail Centers
   - Eases parking for high-traffic areas by guiding visitors to available spots quickly.
2. Airports and Train Stations
   - Provides efficient parking management for frequent turnover and high vehicle volumes.
3. Hospitals and Medical Centers
   - Enhances accessibility and minimizes parking-related delays for staff and patients.
4. Business and Office Complexes
   - Potential to integrate with employee apps to track and reserve spaces in advance.
5. Smart City Initiatives
   - Supports municipal efforts to reduce urban congestion and pollution by optimizing parking.
6. Event Venues and Stadiums
   - Manages peak-time parking efficiently, especially for large-scale events.

Methodology

The project follows an Iterative and Incremental Development approach, starting with the core functionality of detecting occupied and free parking spots using image processing. Features are developed in cycles, tested, and refined, with each iteration improving the system. Additional components like signaling logic and YAML-based storage are integrated progressively. Future enhancements, such as pathfinding algorithms, are planned to ensure adaptability and continuous improvement.

- Week 1-2: Research and Planning: Finalize requirements, tools, and technologies.
- Week 3-5: Drawing Boxes Phase: Develop and test logic to detect parking spaces, drawing bounding boxes around free (green) and occupied (red) slots.
- Week 6-7: Image Processing Implementation: Refine detection algorithms using contours and ensure accuracy.
- Week 8-9: Data Storage Setup: Integrate YAML for parking data management.
- Week 10-12: System Testing and Refinement: Test and optimize the complete system.

Working Model



- Mark Parking Spots: Save coordinates in a YAML file.
- Capture Frames: Use a reference image and real-time feed.
- Image Processing: Blur, clean, and convert to grayscale.
- Difference & Threshold: Compute absolute difference and apply thresholding.
- Detect Contours: Identify contours in thresholded images.
- Object Detection: Use saved coordinates to detect objects.
- Draw Bounding Boxes: Highlight detected objects.
- Classify Spots: Use polygon intersection to mark Occupied or Available.

Algorithms

Image Processing Algorithm:

- Contour Detection (OpenCV): To detect vehicle occupancy in parking spaces, we use contour detection in OpenCV. This involves finding the outlines of objects (cars) within a defined parking space by comparing each frame of the video feed to a reference image. The system identifies contours based on the differences in pixel intensity, which helps in determining if a parking space is occupied.
- Thresholding and Morphological Operations: To reduce noise and enhance the quality of the contour detection, thresholding and morphological operations (erosion and dilation) are applied. These help in isolating significant contours (representing cars) and removing irrelevant background noise.
- Polygon Intersection Algorithm: After detecting contours and bounding rectangles for the vehicles, the algorithm calculates the intersection area between the vehicle's bounding box and the parking spot polygon. If a significant portion of the parking spot is covered by the car, it is marked as occupied.

Data Structures

1. Vectors (std::vector)
   - Description: Dynamic arrays to store elements.
2. Matrix (cv::Mat)
   - Description: Represents images and frames in OpenCV.
3. Point (cv::Point)
   - Description: Represents a 2D point in the image space (x, y coordinates).
4. Moments (cv::Moments)
   - Description: Holds statistical properties of contours, calculates the centroid and area of parking spots.
5. Scalar (cv::Scalar)
   - Description: Represents colors in BGR format.
6. Booleans
   - Description: True/False

SWOT Analysis

Strenghts (S)

Open Source
Scalability
Filling a Vital Need
User-Friendly Interface

Weaknesses (W)

Dependency on Data Quality
Initial Setup Cost
Maintenance Requirements

Opportunities (O)

Mobile App Integration
Potential Partnerships
Navigation Feature Integration

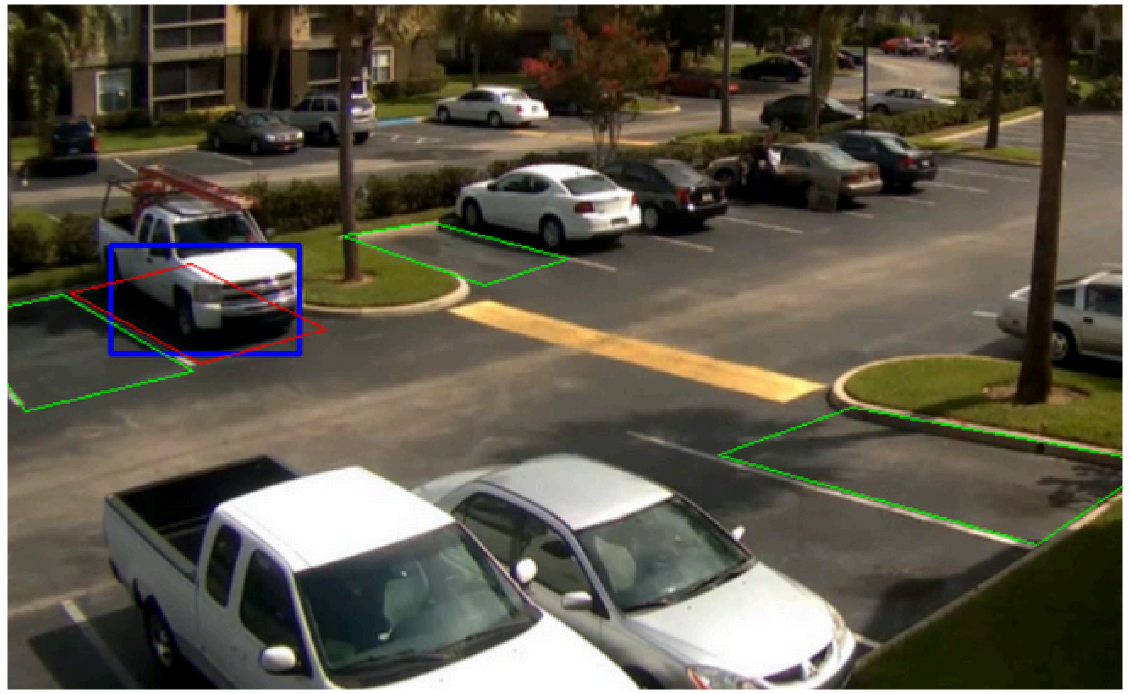
Threats (T)

Competition
Integration Challenges
Technological Limitations

Results

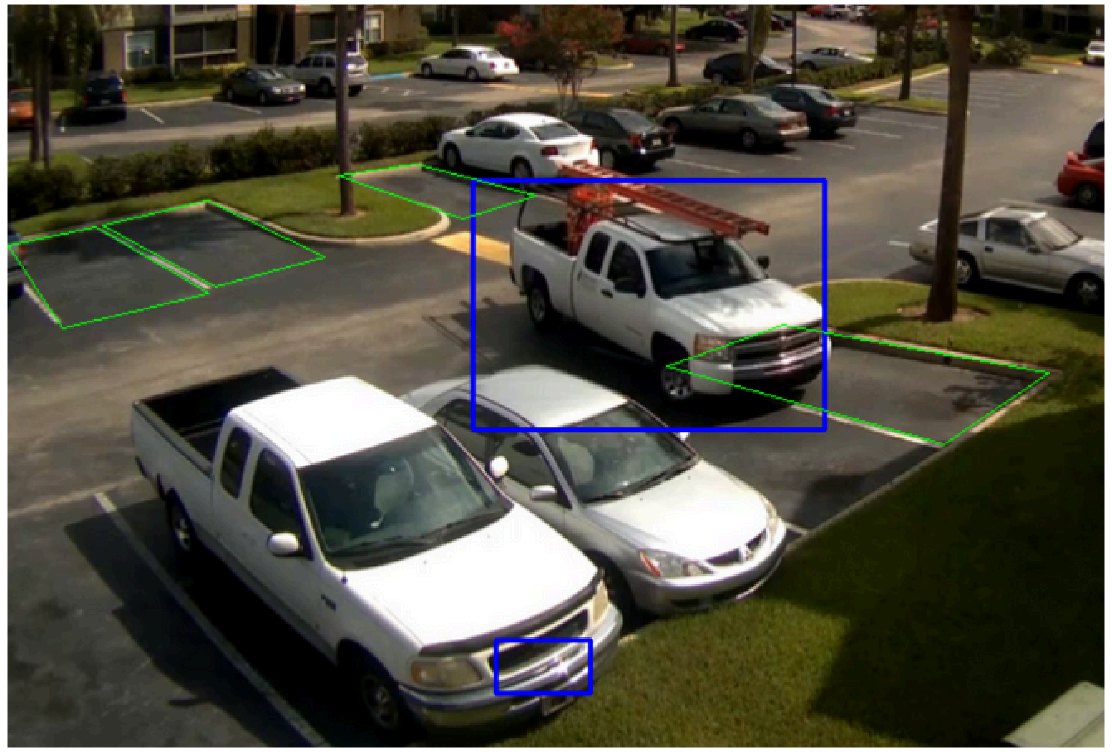
Spot Allotment

Representation of a parking slot changing from available (green) to occupied (red) based on real-time detection.



Drawing boxes

Illustration of the process where the system draws bounding boxes around parking slots to indicate availability.



Contour Detection

Visualization of contour detection and motion analysis to identify occupied and available parking slots

References

[1] O. Rose, "Parking Lot Project," [Online]. Available: https://olgarose.github.io/ParkingLot/. [Accessed: Dec. 12, 2024] OpenCV Documentation. (2024).

[2] D. Acharya and K. Khoshelham, "Real-time image-based parking occupancy detection and automatic parking slot delineation using deep learning: A tutorial," 2020.

[3] Zhao, Y., & Wang, H. (2022). Real-time Image Processing for Smart Parking Systems. IEEE Transactions on Image Processing, 31, 789-800.

[4] OpenCV Tutorials. (2024). Real-Time Image Processing with OpenCV. Retrieved from https://docs.opencv.org/master/d9/df8/tutorial_root.html

[5] Open Source Computer Vision Library. Retrieved from https://opencv.org/releases/

[6] A. Khare, "Automatic Parking Management," GitHub, [Online]. Available: https://github.com/ankit1khare/Automatic-Parking-Management. [Accessed: Dec. 12, 2024].