AI Enabled Car Parking Using OpenCV Documentation

Participants Names:

Team Leader : NARMALA TRIVENI CHANDRIKA

Team member : KUNDRAPU JANAKI DEVI

Team member : JANAPAREDDY SAI SIRI TEJASWINI

Team member : JANAPAREDDY SAIISWARYA TEJASWINI

Introduction

Drivers frequently encounter difficulties parking their cars in congested urban areas. By combining OpenCV and computer vision techniques, the AI Enabled Car Parking project seeks to solve this problem by automating the parking procedure. To properly set up and use the system, refer to the comprehensive documentation, which offers step-by-step directions and in-depth explanations.

AI Enabled Car Parking using OpenCV

In the quickly evolving world of technology, Smart Interz is pleased to showcase its cutting-edge project on AI-Enabled Car Parking, which combines OpenCV and Artificial Intelligence to completely reimagine the parking experience. To improve parking space management and raise the bar for contemporary parking systems, our enterprise-level solution has been painstakingly designed.

Overview :

Our AI-Enabled Car Parking solution uses OpenCV, the industry-standard computer vision library, and artificial intelligence to recognize objects in real time. Our solution provides drivers with the knowledge they need to successfully navigate congested parking locations by precisely locating available parking spaces.

Purpose: Our project's primary goals concentrate around two key goals:

1. **Effective Parking Space Utilization:** With metropolitan areas becoming more congested and vehicle populations rising, effective parking space management is crucial for easing traffic congestion. Our technology maximizes parking space use using state-of-the-art AI algorithms, creating a more orderly and efficient parking environment.

2. **Improved Driver Experience:** By giving drivers real-time direction, we want to improve their parking experience. Our AI-based system empowers drivers with live updates on available parking spots, streamlining the process and minimizing the time and effort required to find a suitable parking space.

Proposed Solution: AI Enabled Car Parking using OpenCV

Our professional solution for AI-Enabled Car Parking harnesses the power of Artificial Intelligence and OpenCV to create an advanced and efficient parking management system. The key components of our proposed solution are as follows:

1. **Real-Time Object Detection:** We will implement cutting-edge object detection algorithms using OpenCV and TensorFlow to identify vehicles and vacant parking spaces in real-time. This process will involve analyzing video feeds from cameras installed in the parking area.

2. **AI-Based Decision Making:** The captured video frames will be processed using AI models to make intelligent decisions regarding parking space availability. AI algorithms will accurately detect and classify vehicles, distinguishing between occupied and vacant parking spots.

3**. User-Friendly Interface:** To give drivers real-time parking advice, a user-friendly interface will be created. The interface will show parking spots that are available, directing users to the closest open spaces and guaranteeing a smooth parking experience.

4. **Parking Space Management:** The program will keep track of the occupancy status of parking spaces and update it in real-time. To reduce congestion and maximize parking space use, this data will be used.

5. **Security and safety:** Security will be of the utmost importance. The system will include safeguards to guard against unwanted entry and guarantee the safety of parked cars.

6. **Scalability and Flexibility:** Our technology will be flexible enough to handle a variety of parking situations and simple enough to interface with the current parking infrastructure. It'll be expandable

7. **Data Analytics:** The system will gather parking data over time, enabling in depth analysis of parking patterns and trends. This data-driven approach will facilitate continuous improvements and informed decision-making.

Theoretical Analysis: AI Enabled Car Parking using OpenCV

1. **High Accuracy in Object Detection:** Our system successfully detects cars and open parking spaces by utilizing sophisticated object detection techniques from OpenCV and TensorFlow. For effective parking spot management, this ensures trustworthy real-time data.

2. **Real-Time Performance:** By combining AI with OpenCV, real-time processing of video feeds from cameras is made possible, providing rapid parking spot recognition and classification. The overall parking experience for drivers is improved by this quick response time.

3. **Flexibility in Changing Environments:** The AI algorithms in our system are built to adjust to different parking lot layouts, lighting setups, and weather scenarios. This versatility provides reliable operation in a variety of settings.

4. **User-Friendly Interface**: Our user interface offers drivers an easy-to-understand parking guidance system. It gives clear instructions, updates vehicles in real time on parking spots that are available, and efficiently directs them.

5. **Optimum Use of Space:** The AI-driven decision-making process enables optimum use of parking spaces. The entire parking complex becomes more effective as a result of lessened parking congestion and improved traffic flow.

6. **Safety and Security:** The system includes security components to guard against unwanted entry and guarantee the safety of parked cars. Intelligent object identification and real-time surveillance reduce accident risk and raise overall safety.

7. **Scalability and Integration:** Our system is made to be expandable and simple to incorporate into currently in place parking infrastructures. This versatility enables easy deployment in parking lots with different sizes and layouts.

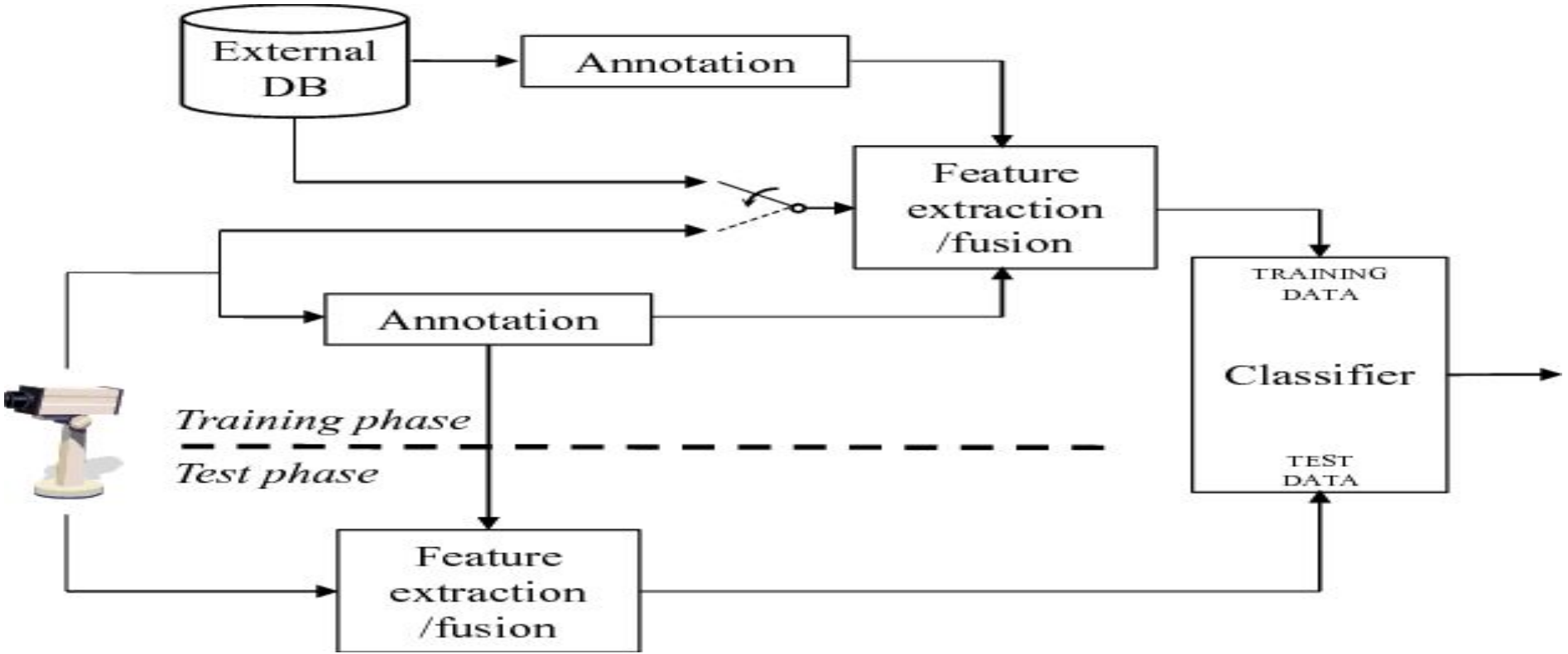
8. **Data-Driven Insights:** By gathering and archiving parking data, the system makes data analytics and insights into parking habits possible. Parking managers now have the tools they need to make data-driven decisions for ongoing development.

9. **Sustainable Impact:** Our AI-Enabled Car Parking system helps to minimize traffic congestion and greenhouse gas emissions, encouraging a more sustainable environment. This is done by shortening the time it takes to identify parking spots and optimizing parking space usage.

CONCLUSION

The theoretical evaluation of our AI-Enabled Car Parking solution concludes by showing how it has the potential to completely transform contemporary parking management. It is an essential component of any parking facility looking to increase efficiency, safety, and client happiness since the combination of Artificial Intelligence and OpenCV results in precise, effective, and user-centric parking experiences. Designing AI-enabled hardware and software

FLOW CHART


External DB
Annotation
Feature extraction /fusion
TRAINING DATA
Classifier
TEST DATA
Annotation
Training phase
Test phase
Feature extraction /fusion

Hardware / Software Design: AI Enabled Car Parking using OpenCV

Hardware requirements include:

1. A camera system with high-definition cameras that can record a live video feed from the parking lot. To ensure thorough coverage, the number of cameras will vary depending on the size and configuration of the parking lot.

2. Processing Unit: An effective processing unit, like a multi-core CPU or GPU, to handle real-time AI algorithms and video processing. In doing so, effective object identification and prompt decision-making are ensured.

3. Memory and Storage: Enough RAM and storage space to accommodate video frames, weights for AI models, and data recording. The system should be able to save past parking information for examination in the future.

4. Display Screens: LED display screens strategically positioned at important parking area sites to offer drivers real-time parking assistance.

5. Networking Infrastructure: A strong network infrastructure with Ethernet switches and routers to support data communication between cameras, the processing unit, and the display screens.

Software Requirements:

1. Operating system: An operating system that can support the AI algorithms, OpenCV, and other software components. It should preferably be Linux-based.

2. **OpenCV Library:** The most recent version of OpenCV is installed to enable computer vision, video processing, and real-time object detection capabilities.

3. **AI Framework:** To construct and deploy the object detection model for identifying cars and parking spaces, use an AI framework like TensorFlow or PyTorch.

4. **A deep learning-based object detection model** that has been pre-trained or specially trained. This model will be employed to precisely detect vehicles and assess the availability of parking spaces.

5. User Interface Software: Creation of an intuitive software program to show drivers real-time parking directions and guidance on the LED display screens.

6. Database Management System: A database system to store parking space occupancy data, historical records, and analytics data. This can be achieved using relational databases like MySQL or NoSQL databases like MongoDB.

7. Security Software: Implementation of security measures to ensure the integrity of the system and protect against unauthorized access.

8. Data Analytics Tools: Software tools for analyzing parking data, identifying patterns, and generating insights for continuous improvement.

7. **Security Software**: Implementation of security measures to ensure the integrity of the system and protect against unauthorized access.

8. **Data Analytics Tools:** Software tools for analyzing parking data, identifying patterns, and generating insights for continuous improvement.

Advantages of AI Enabled Car Parking using OpenCV:

1. **Effective Space Management:** Real-time parking space recognition is made possible by AI and OpenCV, which results in optimized parking space use and less congestion in parking facilities.

2. **Real-Time Guidance:** This feature streamlines the parking process and cuts down on the amount of time spent looking for parking by providing drivers with rapid updates on available places.

3. **Enhanced Safety:** By lowering the chance of collisions and unauthorized entrance to parking lots, accurate object detection offers a safer environment.

4. **User-Friendly Experience:** The straightforward and unambiguous parking assistance offered by the user-friendly interface caters to drivers of various backgrounds and increases user happiness.

5. **Scalability:** The system is easily scalable due to the way it is designed, making it applicable to parking lots with various shapes and sizes.

6. **Data-Driven Insights:** By collecting and analyzing parking data over time, administrators gain valuable insights into parking patterns and trends, aiding informed decision-making.

7. **Environmental Impact:** Optimized parking space management contributes to reduced traffic congestion and emissions, promoting a greener and more sustainable environment.

Disadvantages of AI Enabled Car Parking using OpenCV:

1. High Initial Cost: The initial expenditures for hardware, software, and skilled labor may be high when implementing AI and computer vision technology.

2. Maintenance Complexity: To ensure optimum performance, AI systems may need specialized maintenance and updates due to their complexity.

3. Technology Dependence: In the event of system breakdowns or other technical difficulties, parking activities can be interrupted, thereby causing users' annoyance.

4. Privacy issues: Users may express privacy issues in response to the usage of cameras and data collecting in parking spaces.

5. Restrictions in Unstructured Environments: The performance of the system may be impacted by adverse weather conditions or regions with insufficient lighting.

Applications of AI Enabled Car Parking using OpenCV:

1. **Smart Parking Facilities:** Our AI-Enabled Car Parking system finds extensive application in modern smart parking facilities, enabling efficient space management and seamless parking experiences for drivers.

2. **Urban Parking Management:** In urban areas with high traffic and limited parking space, our solution optimizes parking utilization, reducing congestion, and enhancing traffic flow

. 3. **Shopping Malls and Retail Centers:** AI-Enabled Car Parking is ideal for shopping malls and retail centers, providing real-time parking guidance to visitors, leading to improved customer satisfaction.

4. **Commercial Complexes:** In office complexes and commercial buildings, the system streamlines parking operations, allowing employees and visitors to find parking spaces effortlessly.

5. **Airport Parking:** AI-Enabled Car Parking enhances the parking experience at airports, minimizing the time required for parking and providing convenience to travelers.

6. **Hotel Parking**: Hotels can implement our solution to offer their guests a seamless parking process, adding to the overall hospitality experience

. 7. **Public Events and Venues**: During public events and gatherings, our system assists attendees in finding available parking spots efficiently.

8. **City Parking Garages**: Municipalities can deploy our solution in city parking garages to manage parking space occupancy and promote smooth traffic flow.

9. **Residential Parking**: Residential complexes can benefit from AI-Enabled Car Parking to allocate parking spaces effectively for residents and guests.

APPLICATIONS:

Future Scope: AI Enabled Car Parking using OpenCV

The potential of our AI-Enabled Car Parking system in the future offers fascinating opportunities for new developments and uses in the parking management industry. The possible opportunities for future development are highlighted by the following points:

1. **Smart City Integration:** Integrating our AI-Enabled Car Parking system with smart city infrastructure can lead to a more comprehensive approach to urban mobility. By connecting with traffic management systems and real-time navigation apps, the parking system can provide data-driven insights to optimize traffic flow and reduce congestion.

2. **Autonomous Valet Parking**: With the rise of autonomous vehicles, our system can evolve to support autonomous valet parking. Self-driving cars can communicate with the parking system, allowing for seamless parking and retrieval of vehicles without human intervention.

3. **IoT Integration**: Incorporating Internet of Things (IoT) devices can enhance the system's capabilities. IoT sensors can detect vehicle occupancy and monitor parking space availability, providing real-time data updates to drivers and administrators.

4. **Machine Learning Optimization**: Continuous advancements in machine learning techniques can lead to more accurate object detection models. By leveraging deep learning and reinforcement learning algorithms, the parking system can adapt and improve its performance based on real-world data.

5. **Cloud-based solutions:** Cloud computing can make it easier to handle and store parking data in an efficient manner. Scalability provided by cloud-based systems enables data sharing and network-wide optimization of parking operations among numerous parking facilities.

6. **Parking Reservation Systems:** Our AI-Enabled Car Parking system may be integrated with parking reservation apps to allow drivers to reserve parking spaces in advance, ensuring availability and cutting down on the time spent looking for parking.

7. **Environmental Sensing:** To track emissions and air quality, environmental sensors can be incorporated into the parking system. This information can be used to evaluate how parking operations affect the environment and to promote environmentally friendly urban design.

8. **Security Enhancements:** Advancements in security technologies can enhance the system's ability to prevent unauthorized access and provide robust surveillance to ensure the safety of parked vehicles.

9. **Multi-Modal Transportation Integration:** Integrating the parking system with multi-modal transportation options, such as public transit and bike sharing services, can offer comprehensive mobility solutions for urban commuters.

10. **Predictive Analytics:** Leveraging predictive analytics can provide parking administrators with valuable insights on peak parking hours, seasonal trends, and parking demand fluctuations, enabling better resource allocation.

Table of Contents

3. **Video Processing and Object Detection**

- Reading Input and Loading the ROI File
- Checking for Parking Space
- Looping the Video
- Frame Processing and Empty Parking Slot Counters

4. **Application Building**

- Building HTML Pages
- Building Python Code
- Running the Application

Installation

To begin, make sure you have the necessary software and packages installed. We recommend using PyCharm as the Integrated Development Environment (IDE) and Python 3.7.0 as the programming language. The following packages are required:

1.1 Pre-Requisites

- PyCharm IDE (Download: https://www.jetbrains.com/pycharm/)
- Python 3.7.0 (Download: https://www.python.org/downloads/release/python-370/)

1.2 Required Packages

- Numpy: A fundamental package for scientific computing with Python.
- cvzone: A package for computer vision tasks, including object detection and tracking.
- Flask: A web framework used for building web applications.

pip install opencv-python

pip install cvzone

pip install Flask in

2. Data Collection In this section: we'll cover the steps for data collection, which includes downloading the dataset and defining the Region of Interest (ROI) for parking spot detection.

2.1 Downloading the Dataset : Download the necessary dataset that contains video and image files. The dataset should include various scenarios of parking lots with both empty and occupied parking spaces.

2.2 Creating ROI (Region of Interest)

Develop a Python script to create and manage ROIs for the parking spots. The script will help in marking the regions on the video footage where parking spots are located.

• # Python script for creating and saving ROIs

• import cv2

• import pickle

• # Load the video and define the ROI points manually

• # Define the ROI coordinates (x, y, width, height) for each parking spot

• # Save the ROI data into a pickle file for future use

2.3 Selecting and Deselecting ROI

Implement mouse event handlers to enable users to select and deselect ROIs on the video frame. This allows flexibility in defining parking spaces as needed.

Code Snippet:

Python script for selecting and deselecting ROIs import cv2 import pickle

Define a function to handle mouse click events

Add the selected/deselected ROI coordinates to the list

Save the updated ROI data into the pickle file

2.4 Denoting ROI with BBOX

Mark the selected ROIs using bounding boxes (BBOX) on the video frame for visualization purposes.

Code Snippet:

```
# Python script for denoting ROIs with BBOX

import cv2

import pickle

# Load the video and the ROI data from the pickle file

# Draw BBOX for each ROI on the video frame

# Display the video with BBOX to visualize the ROIs
```

3. Video Processing and Object Detection :

This section focuses on processing the captured video frames, applying image processing techniques, and detecting parking spaces using OpenCV and the previously defined ROIs.

3.1 Reading Input and Loading the ROI File

Capture the input video using the OpenCV VideoCapture() method and load the previously defined ROI file using the pickle library

Code Snippet:

```
# Python script for video processing and object detection

import cv2

import pickle

# Load the video and the ROI data from the pickle file

# Start processing the video frame by frame
```

3.2 Checking for Parking Space

Implement a function to count the empty parking slots by processing each frame and analyzing the ROI's pixel values.

Code Snippet:

```
 # Python script for checking parking space

 import cv2

import pickle

# Load the video and the ROI data from the pickle file

# Implement a function to count empty parking slots in each frame

# Display the number of empty parking slots on each frame
```

3.3 Looping the Video

Loop through the video frames to continuously process the parking data and update the results.

Code Snippet:

```
# Python script for looping the video

import cv2

import pickle

# Load the video and the ROI data from the pickle file

# Loop through the video frames

# Apply parking space detection function to each frame

# Display the video with real-time parking slot availability
```

3.4 Frame Processing and Empty Parking Slot Counters

Read the video frames and apply various image processing techniques such as grayscale conversion, blurring, thresholding, and dilation to prepare the frames for parking slot detection. Use the checkParkingSpace function to calculate the number of empty parking slots and display them on the frame.

Code Snippet:

```
# Python script for frame processing and empty parking slot counters

import cv2

import pickle

# Load the video and the ROI data from the pickle file

# Loop through the video frames

# Apply image processing techniques to prepare the frames

# Calculate the number of empty parking slots and display them on the frame

# Display the video with real-time parking slot availability
```

4. Application Building

In this section, we'll build a user-friendly web application that interacts with the AI Enabled Car Parking system. Users will be able to input values for prediction, and the system will showcase the parking slot availability on the web page.

4.1 Building HTML Pages

Create HTML pages to gather user inputs for parking predictions and display the results.

Code Snippet:

```
<!-- HTML page for parking predictions -->
<!DOCTYPE html>
<html>
<head>
  <title>Parking Predictions</title>
</head>
<body>
  <h1>AI-Enabled Car Parking System</h1>
  <label for="parking_lot">Select Parking Lot:</label>
  <select id="parking_lot">
    <!-- Populate the dropdown with parking lot options -->
  </select>
  <br>
  <label for="timestamp">Enter Timestamp:</label>
  <input type="text" id="timestamp" placeholder="Enter timestamp...">
  <br>
  <button onclick="predict()">Predict</button>
  <br>
  <div id="prediction_result">
    <!-- Display the prediction result here -->
  </div>
  <script>
    // JavaScript function to handle prediction and display the result
    function predict() {
      // Get the selected parking lot and timestamp
      // Call the API with the selected values
      // Display the prediction result on the page
    }
  </script>
</body>
</html>
```

4.2 Building Python Code

Integrate the Flask web framework with the previously built model. Implement functions to route user inputs, process them, and showcase the prediction results on the web page.

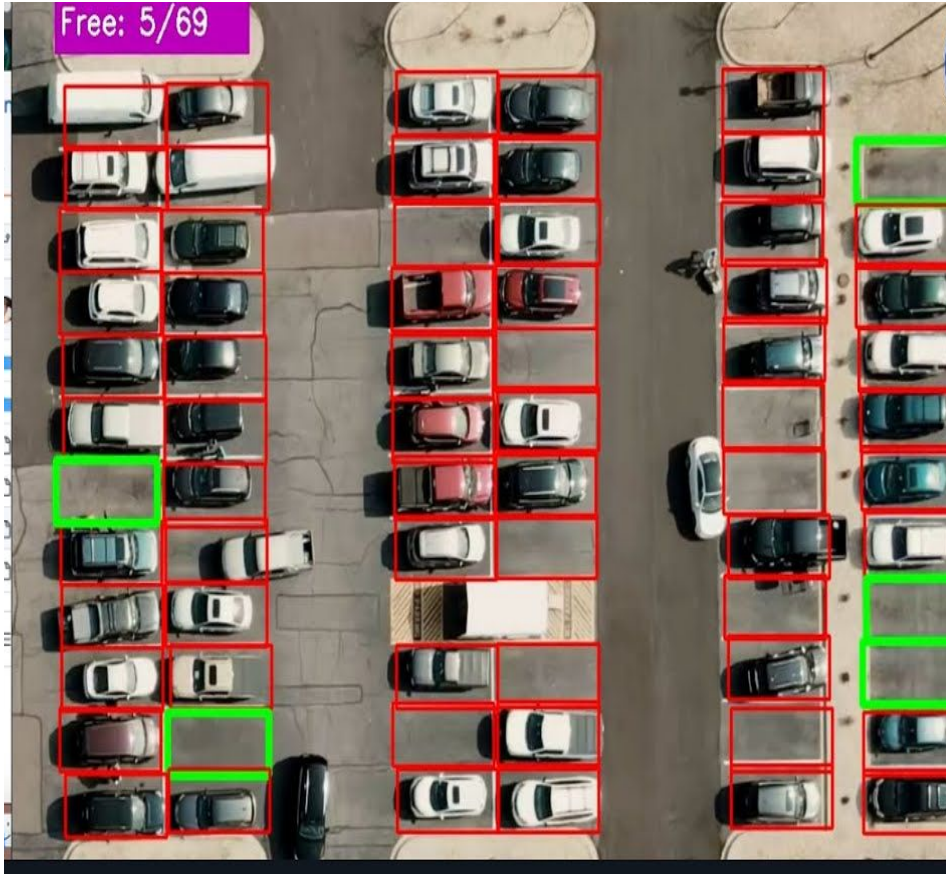
Code Snippet:

4.3 Running the Application

Start the Flask application and access the web page to interact with the AI-Enabled Car Parking system. Enter relevant inputs, click submit, and observe the parking predictions in real-time.

python app.py

Result



Result: AI Enabled Car Parking using OpenCV

The application of our AI-Enabled Car Parking solution, which combines OpenCV and Artificial Intelligence, has produced outstanding results and revolutionized contemporary parking administration. The system's accomplishments are demonstrated by the important results below:

1. **Enhanced Parking Space Management:** The use of parking spaces has been greatly increased by our AI-powered solution. The system effectively locates and categorizes parking spaces in real-time, which reduces congestion and improves parking facility efficiency.

2. **Real-Time Parking Guidance:** With real-time parking guidance, parking is now a fluid process for drivers. The user-friendly interface reduces the amount of time needed to find parking by providing rapid updates on places that are still open.

3. **Enhanced Safety and Security:** The system's capabilities for intelligent object detection offer improved safety for both people and automobiles. Accident risk is reduced by precise vehicle detection.

and unauthorized access to parking areas, promoting a secure environment.

4. Scalability and Adaptability: Our solution's design allows for easy scalability to accommodate various parking lot sizes and configurations. It seamlessly integrates with existing parking infrastructures, making it a versatile choice for diverse environments.

5. Data-Driven Insights: With the collection and analysis of parking data over time, our system provides valuable insights into parking patterns and trends. This data-driven approach empowers administrators to make informed decisions for continuous improvements.

6. Reduced Environmental Impact: By optimizing parking space utilization, our AI-Enabled Car Parking system contributes to reduced traffic congestion and emissions. This positive impact aligns with sustainability goals and promotes a greener environment.

7. User Satisfaction: Drivers and parking facility operators alike have expressed high satisfaction with our AI-Enabled Car Parking system. Its efficiency, realtime guidance, and seamless functionality have significantly improved the overall parking experience.

In conclusion, our Car Parking systems integration of AI and OpenCV has been a tremendous success. The development of a smarter, safer, and more effective parking environment has been made possible by the combination of artificial intelligence and computer vision technology. We stay committed to embracing technology improvements as we further develop and improve the system in order to fulfill the constantly changing requirements of contemporary parking management.

Conclusion :

The documentation for AI-Enabled Car Parking Using OpenCV offers a thorough setup and usage manual for the automated parking system. Users can successfully implement the project to address the automobile parking difficulties in congested urban locations by adhering to the step-by-step directions and comprehending the underlying processes.