

AI-enabled Car Parking using OpenCV

Introduction:

Car parking is a common problem faced by drivers in busy urban areas. The difficulty in finding available parking spots often leads to frustration and anxiety. This project aims to revolutionize the parking experience by automating the process using AI and computer vision technologies. By developing an intelligent system that can identify empty parking spaces, drivers can efficiently locate and secure parking spots without wasting time and effort.

Project Overview:

The AI-enabled car parking system utilizes the OpenCV (Open Source Computer Vision) library and a camera to capture live video footage of the parking lot. The project involves the following key components and skills:

1. **Python:** The project is implemented using Python programming language, which provides a powerful and flexible environment for developing computer vision applications.
2. **Python Web Frameworks:** Python web frameworks such as Flask are used to build the user interface and handle requests from users.
3. **IBM Cloud:** The project leverages the capabilities of IBM Cloud platform to deploy and host the application. IBM Cloud offers scalability, reliability, and various services that enhance the functionality of the system.
4. **IBM Cloudant DB:** IBM Cloudant DB, a NoSQL database service, is utilized to store and manage parking-related data such as the count of available parking spots.
5. **OpenCV:** OpenCV is a widely-used computer vision library that provides various functions and algorithms for image and video processing. In this project, OpenCV is used for object detection to identify empty parking spaces in the live video footage.

6. **Object Detection:** Object detection techniques are employed to detect and locate cars within the captured video frames. This allows the system to determine whether a parking space is occupied or available.

Project Workflow:

The AI-enabled car parking system follows the following workflow:

1. **Video Capture:** The system captures live video footage of the parking lot using a camera connected to the system.
2. **Object Detection:** OpenCV is utilized to perform object detection on the video frames. The algorithm identifies cars within the frames and determines their locations.
3. **Parking Space Identification:** By analyzing the object detection results, the system identifies empty parking spaces by detecting the absence of cars in specific locations.
4. **Parking Space Count:** The system keeps track of the count of available parking spots by continuously monitoring and updating the status of each parking space.
5. **User Interface:** A user-friendly interface is developed using Python web frameworks, allowing users to view the available parking spots in real-time.
6. **Data Management:** The system stores and manages parking-related data, including the count of available parking spots, using IBM Cloudant DB. This ensures data persistence and facilitates efficient retrieval and analysis.

Conclusion:

The AI-enabled car parking system using OpenCV demonstrates the potential of AI and computer vision technologies in revolutionizing the parking process. By automating the identification of empty parking spaces, drivers can save time and reduce frustration when searching for parking. The project combines

Python programming, web frameworks, IBM Cloud services, and OpenCV to create an intelligent system that streamlines the parking experience.

Acknowledgements:

We would like to express our gratitude to SmartInternz platform and SmartBridge organization for providing us with the opportunity to work on this project. Their support and guidance throughout the project were invaluable. We also extend our thanks to the mentors and experts who provided constant assistance and valuable insights, enabling us to overcome challenges and achieve the project's objectives.

Skills Developed:

Throughout the project, the following skills were developed:

1. Proficiency in Python programming language for implementing computer vision algorithms and web application development.
2. Experience in working with Python web frameworks like Flask for building user interfaces and handling user requests.
3. Knowledge of IBM Cloud platform and its services for deployment, hosting, and data management.
4. Understanding of OpenCV library and its capabilities for object detection and image/video processing.
5. Familiarity with AI concepts and their application in real-world scenarios.

References:

- [1] OpenCV: <https://opencv.org/>
- [2] IBM Cloud: <https://www.ibm.com/cloud>
- [3] IBM Cloudant DB: <https://www.ibm.com/cloud/cloudant>
- [4] Python Flask: <https://flask.palletsprojects.com/>