AME 598 : Programming for IoT

Final Project Report

<u>SmartPark – IoT based Parking Solution!!</u>

Faculty

Tejaswi Linge Gowda

Submitted By

Ketan Choudhary and Jay Bhanushali

Submitted On

05 December 2023

1. INTRODUCTION

Parking inefficiencies in urban areas have long been a concern, leading to wasted time, fuel, and increased stress. In response, this project proposes a smart parking solution utilizing Internet of Things (IoT) technology and AWS IoT services. The system aims to provide real-time updates on parking spot availability through a user-friendly interface, enhancing the overall parking experience.

2. PROJECT STATEMENT

The aim of our project is to address the persistent challenges associated with parking in metropolitan cities by developing a Smart Parking System. The conventional approach to parking often results in wasted time, increased fuel consumption, and heightened stress levels due to the difficulty in finding available parking spaces. Our solution involves the creation of a low-cost system utilizing Raspberry Pi and the Pi CAM Module for real-time detection and updating of parking spot availability.

The key objectives of our project include:

1. Efficient Parking Detection: Implement a system that can accurately and efficiently detect available parking spots in real-time.
2. User-Friendly Interface: Develop a mobile app and webpage interface that provides users with instant updates on parking spot availability, making the parking process more convenient.
3. Affordability and Versatility: Design a cost-effective system that can be easily installed at any parking location, ensuring broad applicability across diverse urban landscapes.
4. Traffic Reduction: Mitigate traffic congestion by providing users with timely information about available parking spots, thereby minimizing the need for aimless circling.
5. Environmental Impact: Contribute to the reduction of fuel wastage and pollution by promoting a more efficient use of parking spaces and minimizing the time spent searching for spots.

Through the successful implementation of these objectives, our Smart Parking System aims to enhance the overall parking experience in metropolitan areas, offering a practical and accessible solution to the persistent challenges faced by urban commuters.

3. APPROACH

The project adopts a two-fold approach. First, it involves setting up parking spot positions through computer vision and user interaction. Second, it employs real-time detection techniques using recorded videos to determine parking spot availability. AWS IoT services facilitate seamless communication between the laptop and the cloud, enhancing the scalability and adaptability of the system.

4. METHODOLOGY

The methodology for this smart parking project involves a comprehensive approach to real-time parking spot detection and status updates. Beginning with image capture using a laptop running VS Code, bounding boxes are assigned to each parking spot, enhancing the accuracy of subsequent analyses. Parking activities are then recorded through smartphone videos, simulating real-world scenarios. Computer vision algorithms are employed to dynamically assess parking spot occupancy within the recorded videos, and the system dynamically updates the color of bounding boxes—green for available spots and red for occupied ones. Integration with AWS IoT services facilitates secure cloud communication, enabling the transmission of real-time parking spot statuses. The final step includes the development of a user-friendly interface, accessible through a mobile app or webpage, allowing users to remotely monitor parking availability and make informed decisions. Challenges in accurate bounding box assignment and robust video analysis algorithms will be addressed iteratively, aiming to optimize the system for diverse parking scenarios using readily available resources.

5. SETUP

The project was divided into two parts: hardware setup and software setup. The independent methodologies used greatly aided in the timely development, deployment, and debugging, culminating in the project's successful integration and completion.

a) Hardware Setup: For the hardware setup, begin by selecting IoT devices equipped with the necessary sensors and connectivity, such as the Raspberry Pi and Pi CAM Module for budget-friendly parking spot detection. Connectivity modules like Wi-Fi, and a stable power supply to ensure uninterrupted operation.

b) Software Setup: On the software side, start by installing an operating system compatible with the IoT devices. Configure devices to connect to AWS EC2 for seamless data transmission and displaying the information in a simple format thereby allowing users to access real-time parking information. Thoroughly test the entire system, deploy devices at parking locations, and document software configurations and code for future maintenance.

6. CHALLENGES

Implementing a smart parking system using IoT and AWS IoT presents several challenges, ranging from technical intricacies to logistical considerations. We faced a bunch of challenges with the hardware we initially chose, the Raspberry Pi Zero. It was giving us a lot of delays and slow responses when we tried to use it. So, we switched to the Raspberry Pi 3, which helped us capture photos and videos in real-time, but we couldn't install a crucial library called OpenCV due to some hardware issues. The Raspberry Pi 3 also had problems with overheating, making it unresponsive at times. Because of these troubles and limited resources, we had to change our approach. Instead of using the Raspberry Pi Zero or 3, we decided to do our car parking detection on a laptop using Spyder and VS Code, which turned out to be a more practical solution for us. Some of the key challenges include:

1. Connectivity Issues: Ensuring reliable connectivity between IoT devices and AWS IoT in diverse urban environments can be challenging. Issues like network congestion, signal interference, or device disconnections need to be addressed for continuous and real-time communication.

2. Security Concerns: Security is paramount in IoT systems. Safeguarding devices, data transmission, and user information from potential cyber threats requires robust encryption, secure key management, and adherence to best security practices to prevent unauthorized access or data breaches.

3. Scalability: As the number of connected devices increases, ensuring the scalability of the system becomes crucial. Designing an architecture that can handle a large volume of devices without compromising performance is a complex task.

4. Power Consumption: IoT devices often operate on limited power sources. Optimizing the power consumption of devices, especially in scenarios where continuous operation is essential, poses a challenge that requires efficient hardware design and power management strategies.

5. Real-time Data Processing: Processing and analyzing real-time data generated by multiple devices can strain system resources. Designing an efficient data processing pipeline and implementing effective algorithms for parking spot detection and updates without introducing delays is a significant challenge.

What is the design gap? What you set out to do vs. what you ended up ACTUALLY doing?
The design gap in this project is the disparity between the initial plan and the practical implementation of the hardware and software setup. Initially, the project aimed to use the Raspberry Pi Zero and later the Raspberry Pi 3 with their respective camera modules for real-time car parking detection. However, the chosen hardware posed challenges, including lag issues, overheating, and the inability to install critical libraries like OpenCV. As a result, the project deviated from the original plan and shifted to using Spyder and VS Code on a laptop for car parking detection, abandoning the Raspberry Pi hardware. The design gap, therefore, lies in the shift from the planned Raspberry Pi-based setup to the actual implementation using alternative tools on a laptop due to unforeseen hardware constraints and limitations.

7. RESULTS

Systematic implementation of algorithm development and hardware configuration led to effective use of the demo kit. The screenshot given below shows the system in active state.

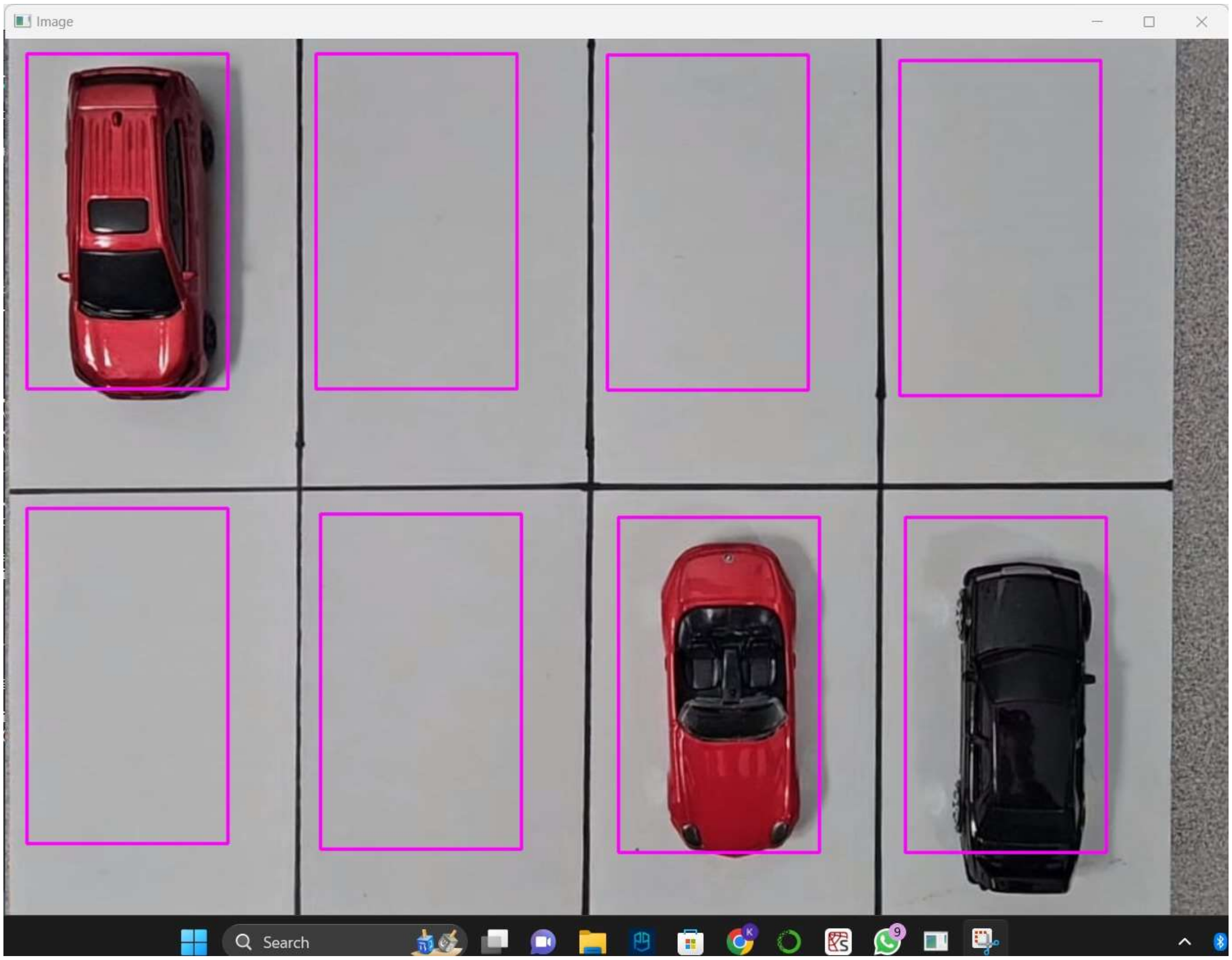

Fig 1. Bounding Boxes

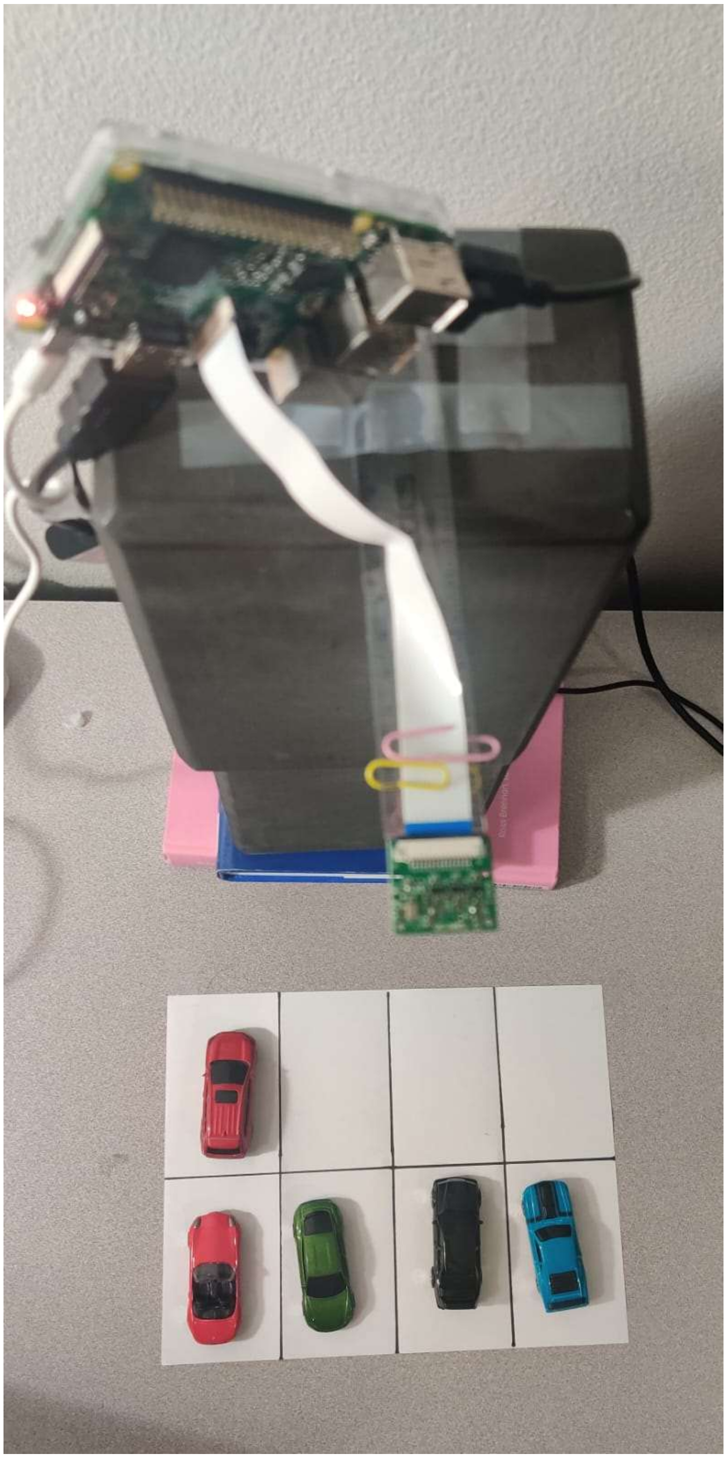

Fig 2. Setup

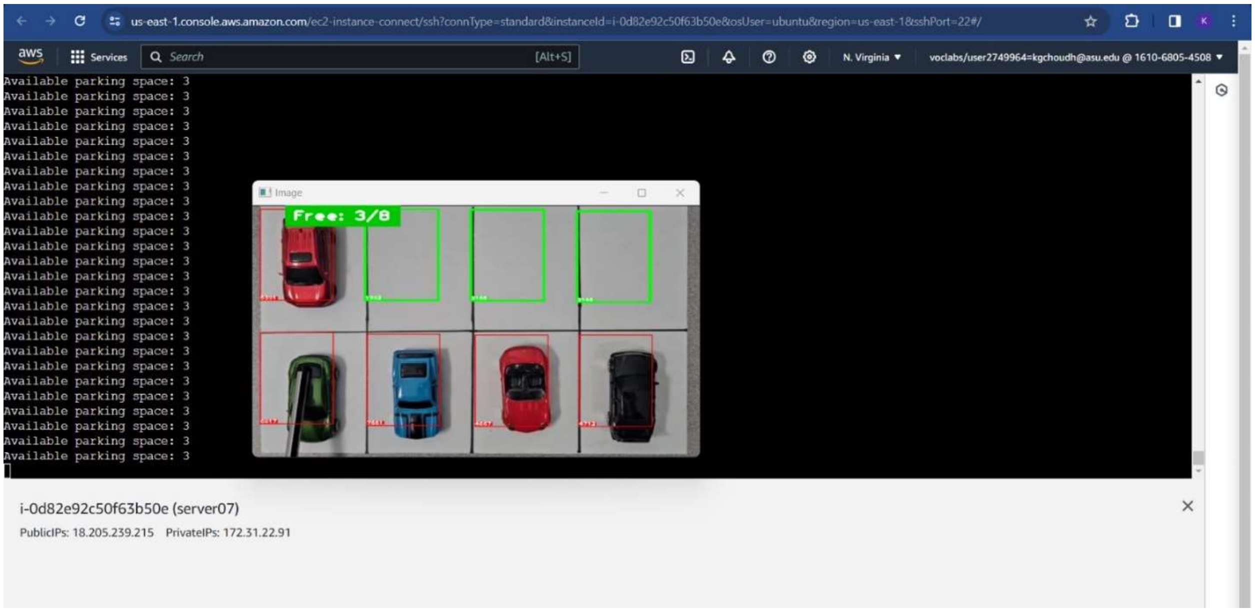

Fig 3.1. AWS EC2 receiving message in real time

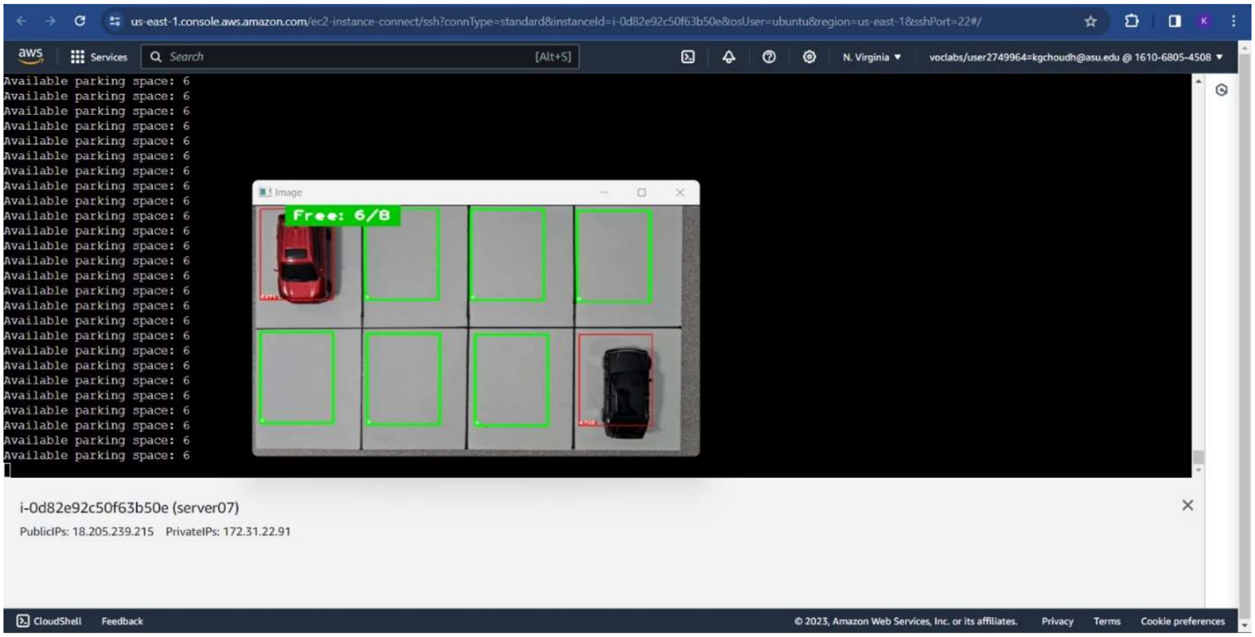

Fig 3.2. AWS EC2 receiving message in real time

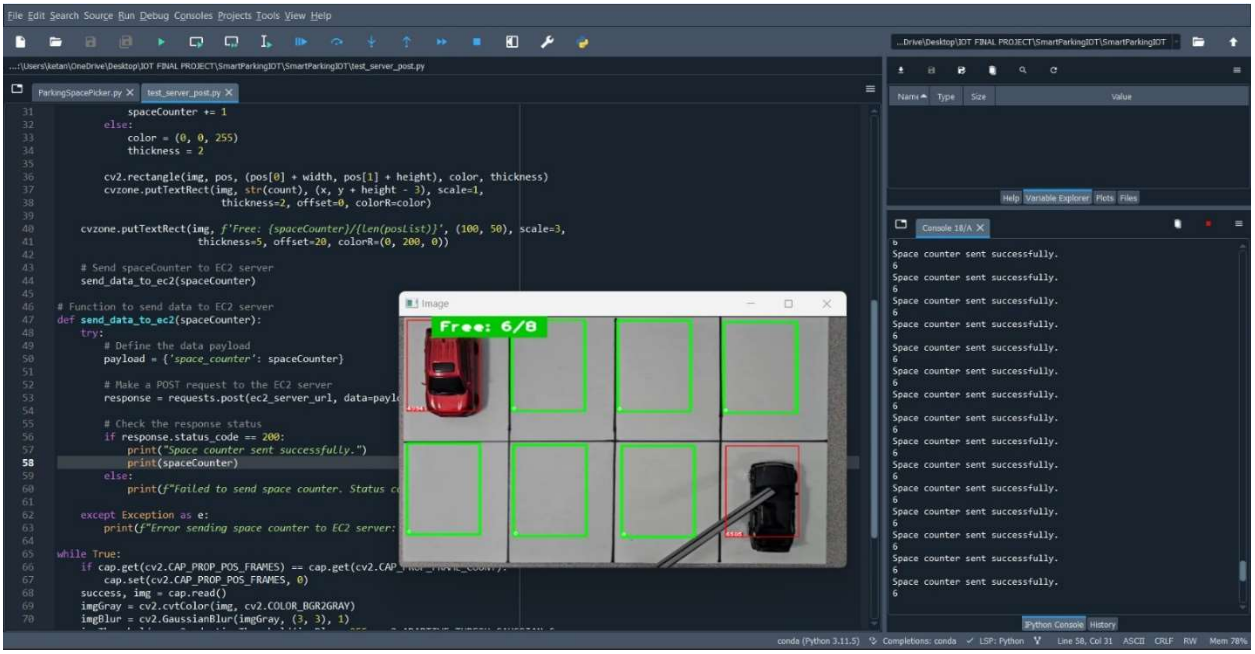

Fig 4. Python Script Successfully running

8. CONCLUSION

The development and implementation of a smart parking system using IoT and AWS IoT present both opportunities and challenges. While the project aims to address the persistent issues of parking in metropolitan areas through real-time detection and user-friendly interfaces, challenges such as connectivity issues, security concerns, and unforeseen complexities in integration may impact the actual realization of the intended design. Bridging the design gap requires a holistic and adaptive approach, incorporating lessons learned during the project lifecycle.