**Report**

**Project Title:** Parking Space Detection and Occupancy Monitoring System

**Objective:** The goal of this project is to develop a real-time system to detect and monitor parking space occupancy using video analysis. By leveraging computer vision techniques, this system will identify vacant parking spots swiftly, reducing the time users spend searching for parking and enhancing overall efficiency.

**Overview of Project Progress**

The project’s primary focus is to implement a system that uses computer vision to monitor parking spaces in real-time. This report outlines the progress achieved so far, specifically on the initial stages of the system's implementation. The project is broken down into seven implementation steps. For today’s deliverable, we have successfully completed the first two steps, which form the foundation of the parking space detection system. These initial steps involve setting up the system environment to handle video feeds and defining parking spaces within the frame, allowing future steps to focus on real-time analysis and detection.

**Implementation Progress**

**Step 1: Setup and Load Video Feed**

The initial step involved setting up the video feed, which will serve as the primary input for the system. This setup is critical as the video feed is the data source for the entire detection and monitoring process. To ensure compatibility and flexibility, we opted to use OpenCV, a versatile computer vision library that supports real-time operations. Using OpenCV, we established a consistent pipeline for capturing video from a live camera feed or a pre-recorded video file.

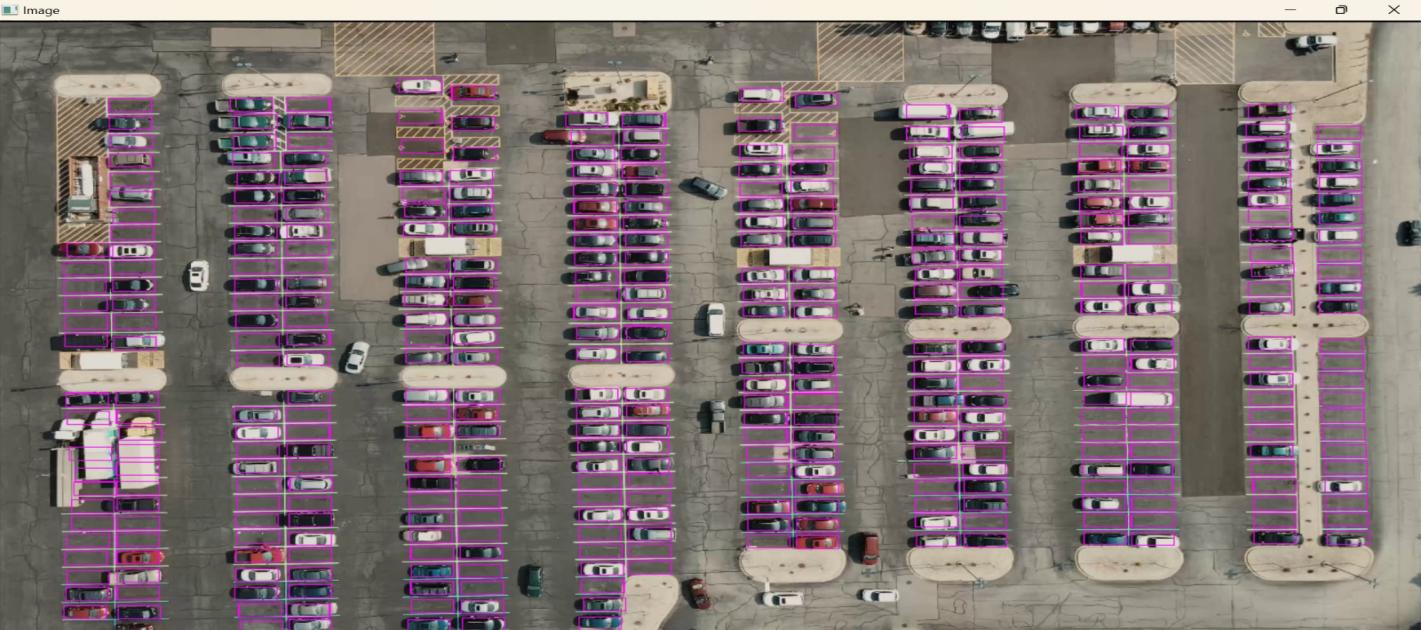
This step required configuring the video capture settings, such as resolution and frame rate, to meet the requirements of real-time analysis. We tested the feed with various video qualities to determine the optimal settings for balancing processing speed with image clarity. A significant portion of this step also included error handling to manage scenarios where the video feed might be interrupted or when frames are dropped. At the end of this phase, we were able to successfully load and display the video feed, confirming that the data input pipeline was operational.

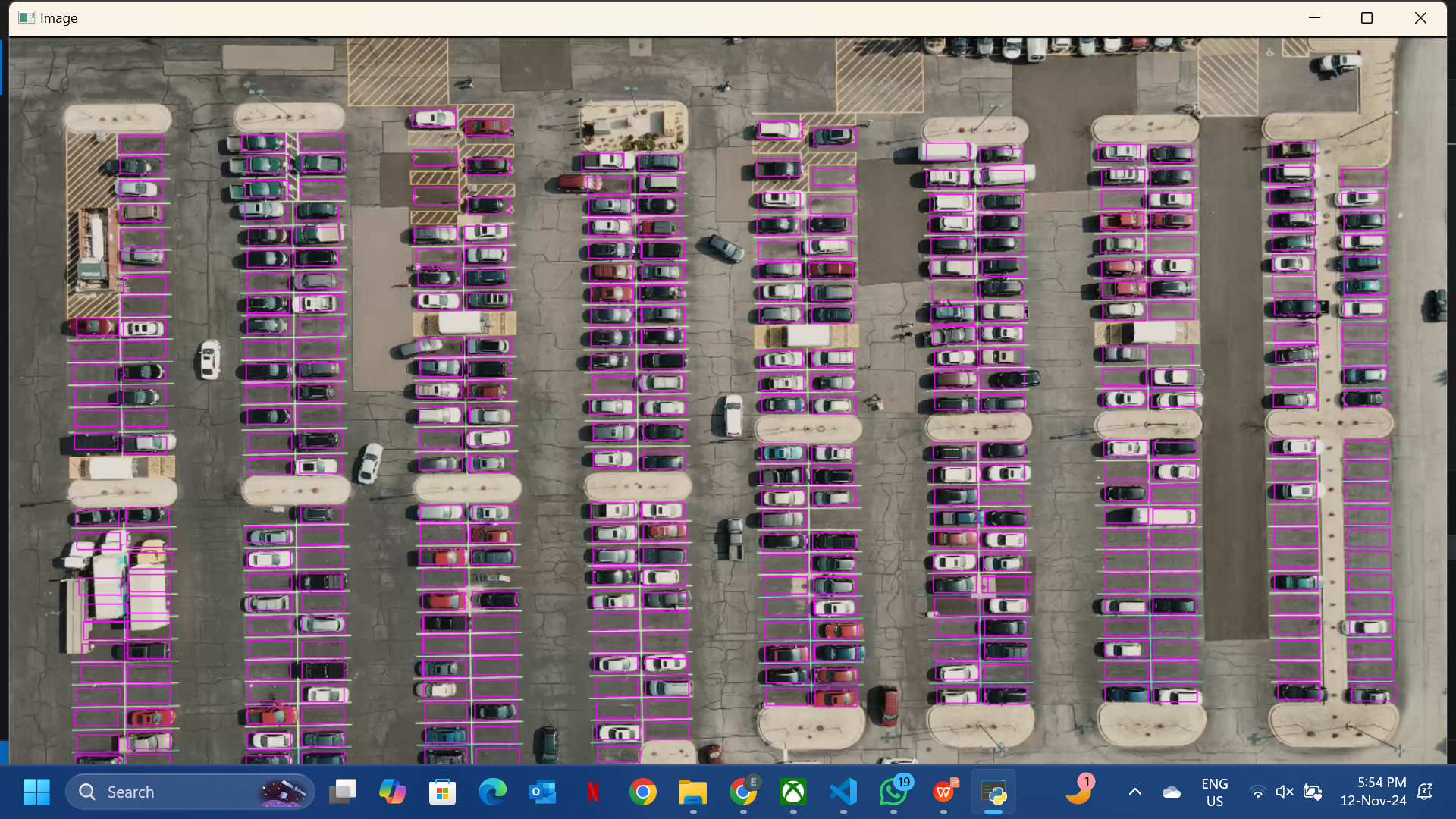
**Step 2: Load and Define Parking Positions**

With the video feed setup in place, the next essential step was to load and define the specific parking spaces that the system would monitor for occupancy. Defining these spaces accurately is critical to achieving reliable detection and minimizing false positives or negatives.

We began by identifying each parking space within the video frame and assigning unique identifiers to each one. This required overlaying a grid on the video feed and manually marking the boundaries of each parking spot. Using OpenCV’s drawing functionalities, we defined these boundaries and saved them as a set of coordinates. Each coordinate set corresponds to a parking space, allowing the system to refer back to these coordinates in later stages for occupancy monitoring.

In addition to defining the parking spots, we also created a configuration file to store these coordinates. This configuration enables flexibility in reusing or modifying parking definitions without altering the main codebase. After loading the defined parking positions, we tested the system’s ability to accurately display each designated spot on the video feed. This test confirmed that the system could accurately recognize each space, marking a successful completion of Step 2.





With Steps 1 and 2 completed, the project is on track for the upcoming phases, which focus on video frame preprocessing, parking space occupancy detection, and occupancy tracking. Each of these steps is crucial for achieving real-time monitoring, which is central to our objective.

Step 3: Preprocess Video Frames – In this step, we will preprocess each video frame to enhance the detection process. This includes resizing frames, converting them to grayscale, and applying filters to improve the system’s ability to differentiate between occupied and vacant spaces.

Step 4: Parking Space Occupancy Detection – Once preprocessing is complete, we will implement the core occupancy detection logic. This involves using computer vision techniques to identify changes within predefined parking spaces and determine whether they are occupied or empty.

Step 5: Track Occupancy Time and Detect Overstays – We will then implement a tracking feature that monitors the duration of occupancy for each space, allowing the system to identify overstays and notify users accordingly.

**Reflections on Progress**

The progress achieved thus far provides a strong foundation for further development. Completing Steps 1 and 2 has confirmed the feasibility of real-time video feed handling and accurate parking space definition, addressing initial concerns regarding compatibility and scalability. While challenges have arisen, they were successfully managed by adapting tools and fine-tuning parameters, which will contribute to the overall robustness of the system.

**Conclusion**

In summary, we have made significant progress on the Parking Space Detection and Occupancy Monitoring System, successfully completing the initial setup and defining parking positions. These achievements position us well to proceed with preprocessing and occupancy detection, which will enable the system to offer real-time monitoring and improve parking efficiency for users. Moving forward, we will continue to document each implementation step and address any technical challenges, ensuring that the project stays aligned with the intended objectives.