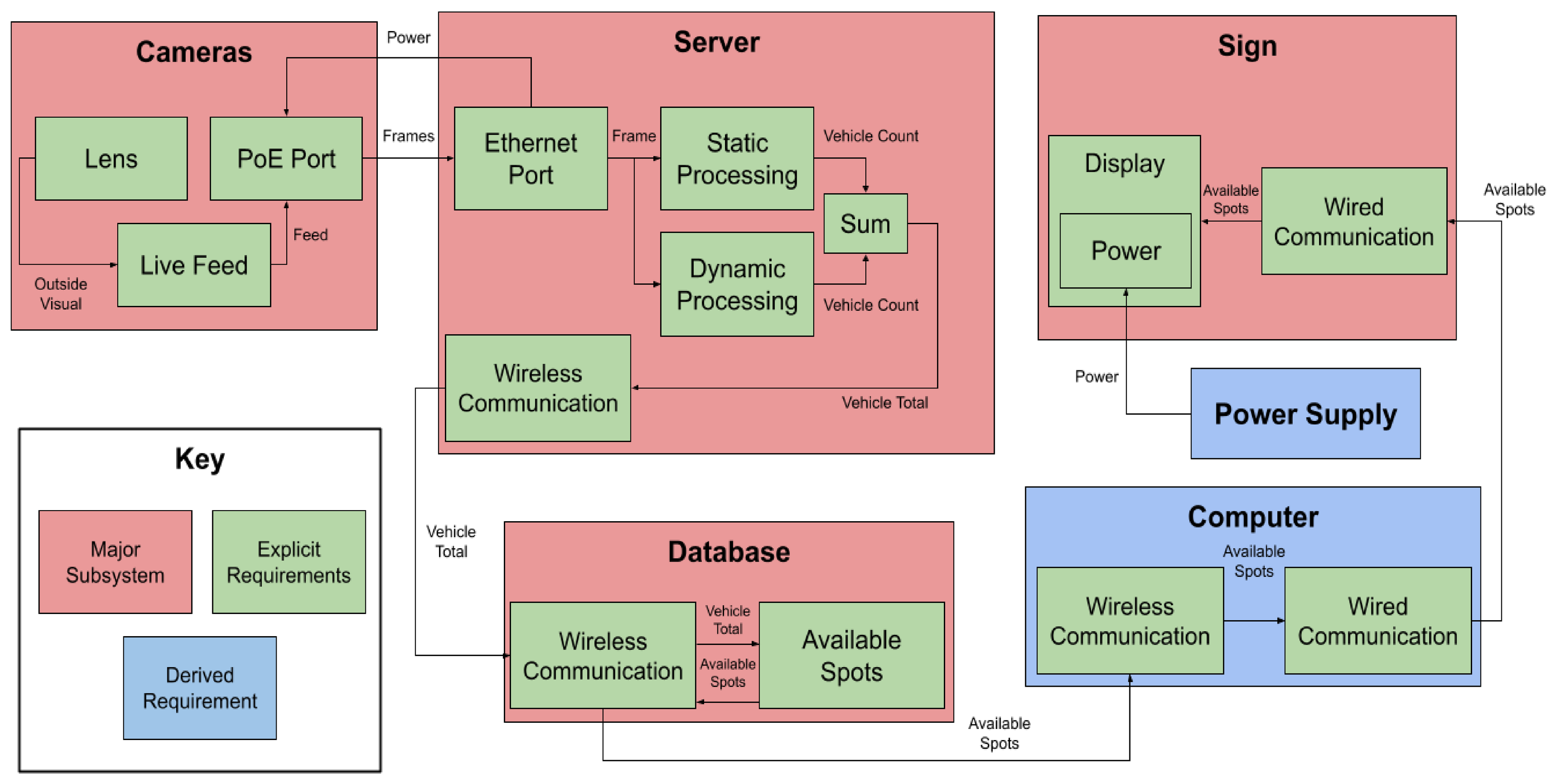
Parking Lot Monitoring System

Kester Nucum Gabriel Laboy Aaron Wilhite Regulo Garza Genevieve Schreiber

Abstract

Students commuting to Tennessee Tech have difficulty finding open parking spots during busy hours of the school day, which causes tardiness to class. To solve this problem, our team's goal was to design a proof of concept for a parking lot monitoring system that uses visual sensors that determine the amount of open spots to be displayed in real-time on a downloadable mobile application and an outdoor sign.

System Design Overview



Cameras - The cameras provide images of parking lot sectors. Each sector is observed with one of two tracking methods: static and dynamic. **Static tracking** cameras monitor parked cars while **dynamic tracking** cameras monitor cars entering and exiting entrances.

Server - The server **receives frames** from the cameras, **processes the frames through the AI models**, computes the sum of vehicles in the parking lot counted from all running models, and **sends the total** to the cloud-based database via Internet.

Sign - The outdoor sign displays the available number of red parking spots to incoming commuters. It can be programmed to **display the number of available parking spots** using an Arduino that is serially connected to an Internet-connected computer that can query the available parking count from the remote database.

Remote Database - The remote database, which was built with Google Firebase by the Computer Science team, **stores processed and static data** for display to the sign and mobile app.

Analysis of AI Processing

Figure 1. Static Overlay

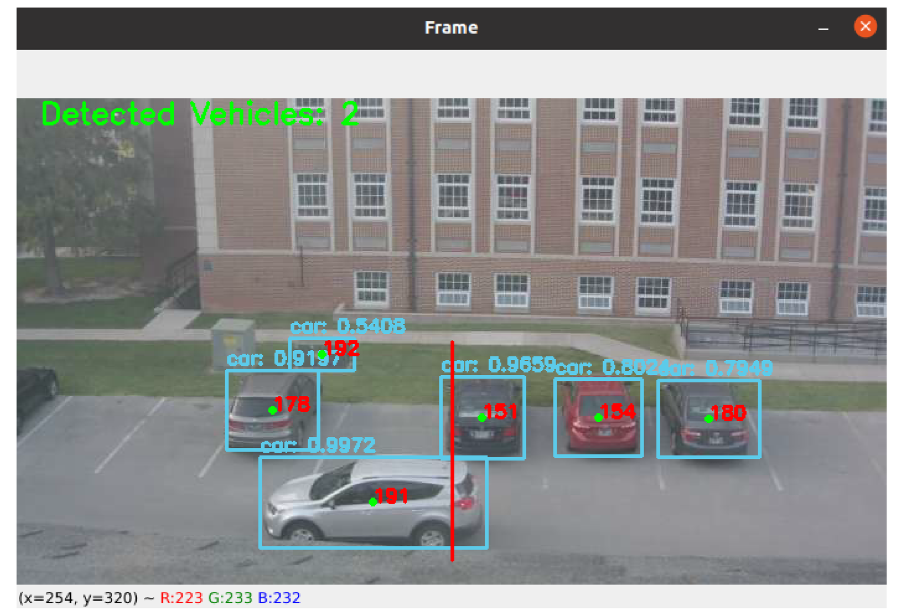


Figure 2. Dynamic Overlay with Vehicle

Static Tracking Model: The static tracking model aims to count the number of cars visible in the frame. Experiment results showed that the model was **more accurate during the afternoon than during morning and midday on a sunny day**. The AI model was also **more accurate in overcast conditions versus clear conditions**. The two major reasons for this occurrence are that too much light in the frame can prevent the algorithm from seeing cars' distinct edges and that YOLOv3 could be biased toward cars in overcast weather conditions. **Our model does not meet our intended accuracy of ±1 car within the actual truth.**

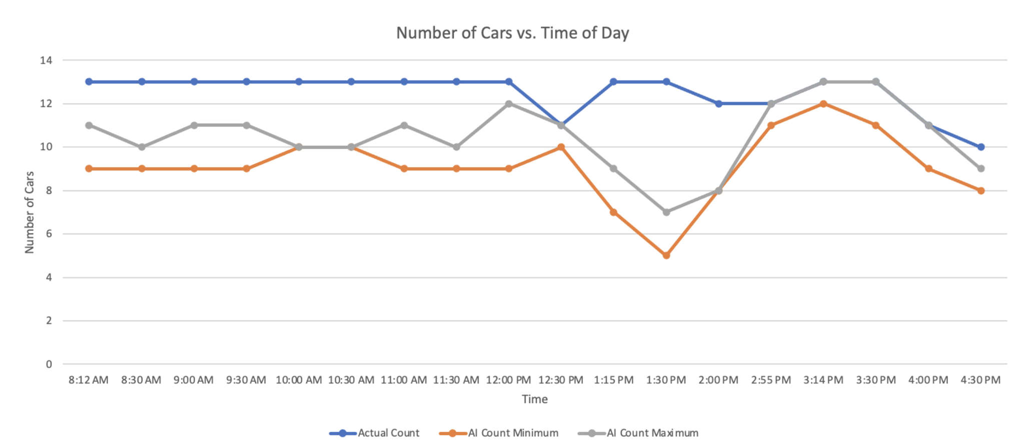


Figure 3. Static AI Vehicle Detection

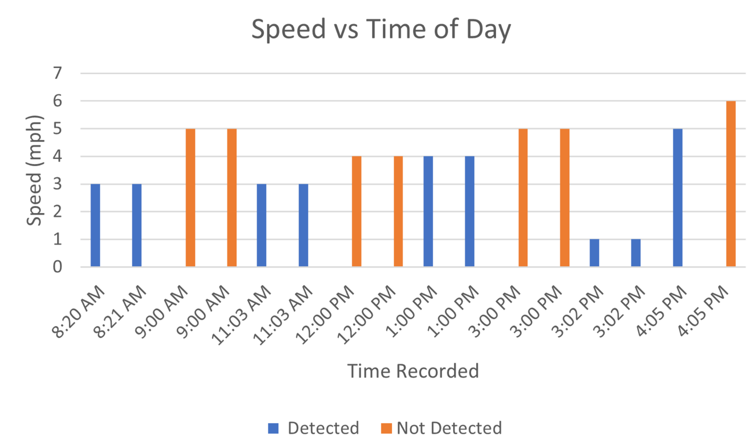


Figure 4. Dynamic AI Vehicle Detection

Dynamic Tracking Model: The dynamic tracking model tracks if a car has crossed a virtually drawn line by comparing its position in sequential frames. However, it can **detect cars crossing the virtual line at no more than 3 mph** due to the slow processing rate of the server. The YOLOv3 algorithm assigns a unique numeric ID to each car in a frame, but if the car travels too far between frames, the algorithm will assign it a different ID. The server did not have a NVIDIA GPU to enable CUDA computations, so the YOLOv3 algorithm was run purely on the CPU, which produced a processing rate of 2 frames per second (fps). **Therefore, our model is functional only for very slow-moving vehicles.**

The Team

Depicted (from left to right): Gabriel, Aaron, Kester, Regulo, Genevieve

Project Conclusions

The project has served as a **proof of concept** in that it is feasible to implement a system that can monitor school parking lots using **only cameras**. By utilizing AI algorithms, serial communications, and data storage on a remote database and building a mobile application, it is possible to give Tech students a remote method of determining the availability of parking spots. The next step to achieve our project's **long term goal is implement the system to monitor an entire lot**.

Future Work - Moving Toward Outdoor Implementation

- Upgrade server to run CUDA-enabled YOLOv3 on NVIDIA GPU.
- Weatherproof and electrically secure sign for outdoor use.
- Implement wireless communication between cameras, server, sign, and remote database.
- Design a solar-based energy source for cameras and sign.
- Install safety casing over cameras.

Acknowledgements

Special thanks to our advisor, **Jesse Roberts**, and our professional mentors at **NAVSEA**, **Rob Sexton and Kristen Voyles**, for their guidance throughout this entire project, as well as to the **Computer Science team** and **Mechanical Engineering team** for their work and collaboration. Thanks to **ITS**, **Parking Transportation**, **University Police**, and **Facilities** for their help as well.