

Software engineering internship at Planon

Fixing car detection Program

Jul 2024 - Aug 2024

During my first internship with Planon, I was tasked with fixing the car detection program which used a Neural Network which analyzed live feed from the raspberry-pi camera to determine whether a car was positioned in the parking spot or not. The live feed analysis was displayed on Planon's server to be accessed by other Planon employees that wanted to know which parking spots were free.



Live analysis of parking displayed onto server



RaspberryPi camera

When I started working the live feed analysis did not function, instead a timeout error was shown on the server.

I was tasked with fixing the program so that the parking lot would be shown from the server including free and occupied spots.

In order to find the error, I needed to familiarize myself with the program.

The program was primarily in C++ with the neural network section being in Python and the uploading onto the server side in NodeJs.

To familiarize myself with the program, I made a memory map to understand how each program spoke to each other, and where memory was being called or pushed. While analyzing the program I tried to separate the flow into different sections: initializing the single frame analysis of the Raspberry-pi video feed, the storing of the frame which was then passed into the neural network to be analysed, another storing section of the analyzed video and the final section responsible with pushing the analyzed frame onto the company's server to then be viewed on a display panel.

Once I separated the flow into different sections, I went on to write tests to find at what point in the flow did the program break.

First I checked if a random image that was not pulled from the video feed would manage to output an analyzed image to test the neural network section, or if a still frame that hadn't been analysed could still be pushed onto the server to check the pushing onto the server section.

When the program passed both of those tests I determined that the issue did not come from the neural network, or the server uploading. I went on to test if the camera was functioning, I found that an image was being correctly initialized as then allocated into a file to be used, however, the file was never correctly cleaned after being used. Instead of saving a singular frame, more memory would be allocated into the same file, causing the analysis of the file to fail.

Based on the failed test case I tried finding ways to try and fix the memory cycle. To do so, I created a destructor, which deallocated the stored memory from the video feed after being pulled by the neural network. This allowed for the neural network to be able to continuously call a file which properly stored the video feed without corrupting it with more memory which it could not analyse. Once I managed to solve the memory storage problem, the neural network went back to functioning properly and displaying the live parking feed onto a display panel which was shown to investors as part of the technology displayed at Planon.