**Friday, September 22, 2023**

One of my deliverables is inventorying the parts we currently have available so that I can determine what is needed to get the car working and send a list of parts needed. I did that for the first half of class and put my results in a spreadsheet. Then, I continued working on transfer learning. I re-wrote my training script using the Keras Applications API I mentioned last week. However, it ran into an out-of-memory exception on my personal computer’s GPU (which has 12 GB of built-in RAM). I fixed the issue by decreasing batch size, although this made training much slower. Currently I am training with the EfficientNetV2B1 model and a Dense layer going to a single output which is the steering angle of the car. First, I froze the EfficientNet layers and then trained only the Dense layer. Then, I unfroze the entire model and trained it across the whole model with a lower learning rate as to not overfit.

**Monday, September 25, 2023**

I drafted and sent the email to Dr. Gabor about the parts I needed today. This deliverable is due on Wednesday the 27th. The main concerns I had were getting high-quality telemetry data from the car’s operation, such as position, velocity, and angle. I determined a sensored steering servo would be needed to know the actual angle of the servo as opposed to the angle demanded by software. Furthermore, I researched the capabilities of the Nvidia Jetson boards and what would be needed to run reinforcement learning locally on-device. I found that both the Jetson boards we have available are somewhat outdated and the newer Jetson Orin series of single-board computers would be more suitable for running reinforcement learning on a continuous image stream given their higher matrix processing throughput. There were also some basic components, such as bumpers and batteries needed to get the car working safely. Furthermore, I anticipate possibly needing a better localization system for the car to run reinforcement learning training on-line in order to reset the car’s position for successive training episodes. I am not sure if the current sensors (IMU, Lidar, wheel odometry, steering odometry) will be sufficient for this. It’s possible I might need something like an ultrawideband (UWB) system to more accurately and consistently localize the car. A Lidar+IMU setup suggests using a SLAM (Simultaneous Localization and Mapping) setup, which uses the Iterative Closest Point algorithm to match features from time-differentiated scans. The IMU data can be used to corroborate the output of ICP.

**Wednesday, September 27, 2023**

I wasn’t able to achieve good results with my own dataset/simulation setup, so I switched to using the Comma.ai research dataset: <https://github.com/commaai/research>. I tried to access the TJ cluster both through SSH and JupyterHub but was unsuccessful. I finally was able to get in through the remote access server, so I could download the dataset but not actually run any code. While the dataset was downloaded, I looked at other strategies for end-to-end control with transfer learning. I found that most approaches use an LSTM or sequence-based approach in which the input to the model is several images concatenated together.

Also, Dr. Gabor asked me to compile the names, prices and buying links for the parts I need so I worked on that towards the end of the period.