Using GPU Computing To Develop a Neural Network Model for Autonomous Driving

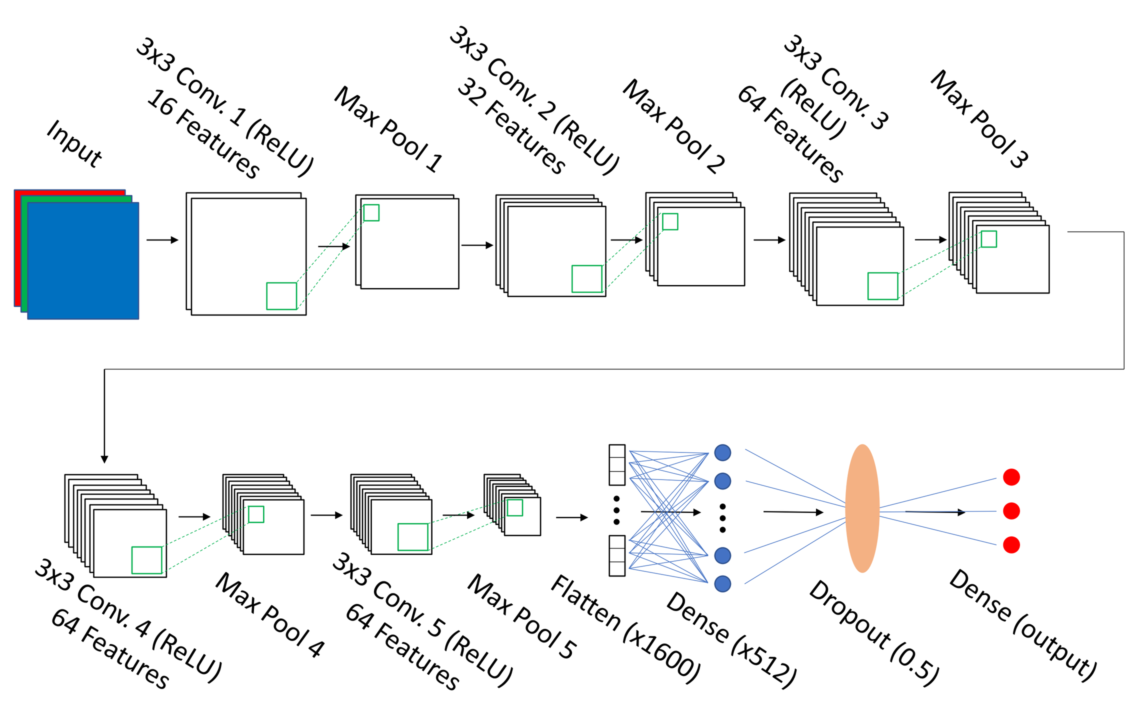
John McCartney

Mentors: Dr. Shadi Alawneh, Karam AbuGhalieh

Introduction 1

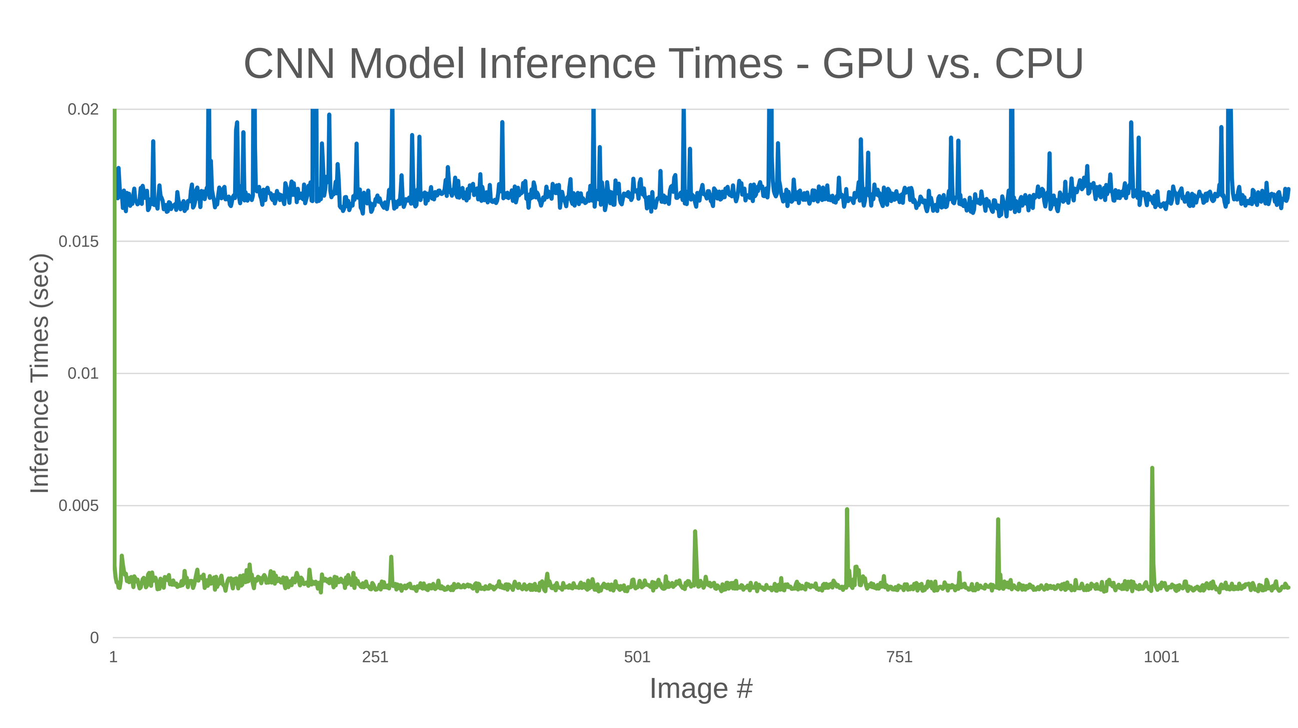
With self-driving cars being the future of the auto industry, the formulation of more efficient algorithms has been the forefront of the advancement of self-driving technologies. Driving scenarios are becoming increasingly complex, so it is crucial that models used for self-driving cars can respond to these scenarios as quickly as possible to ensure maximum safety.

Design 3



For this project, I designed a 14-layer convolutional neural network using the Keras library. There are 5 sets of convolutions and pooling layers, along with a dropout layer. The output are 3 neurons which represent 3 directions (left, right, forward). The robot will drive in the direction corresponding to the direction that is output. I utilized the Google Colaboratory software to develop and train my model. This made it very simple to use GPUs and CPUs to run my code.

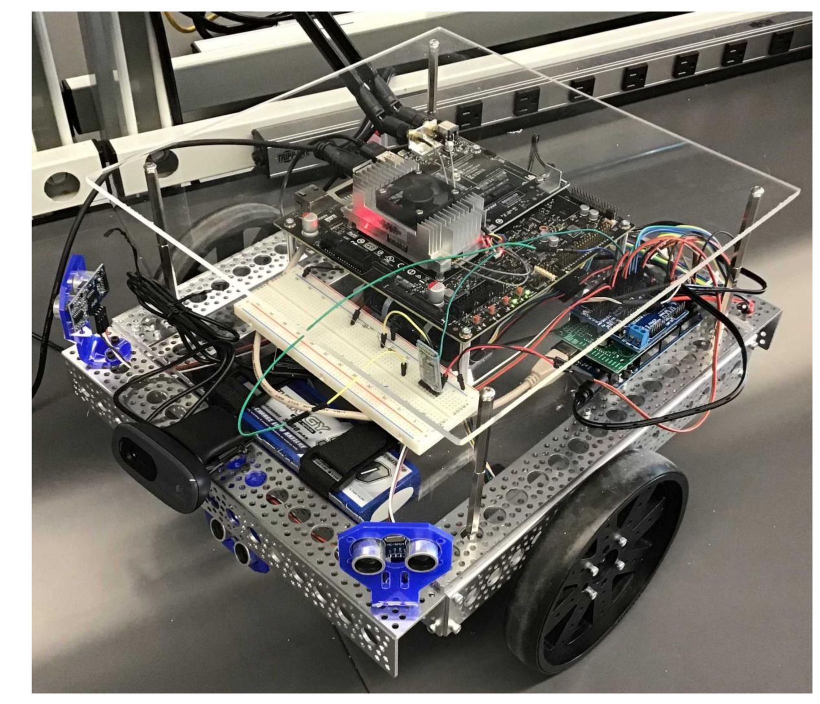
Inference 5



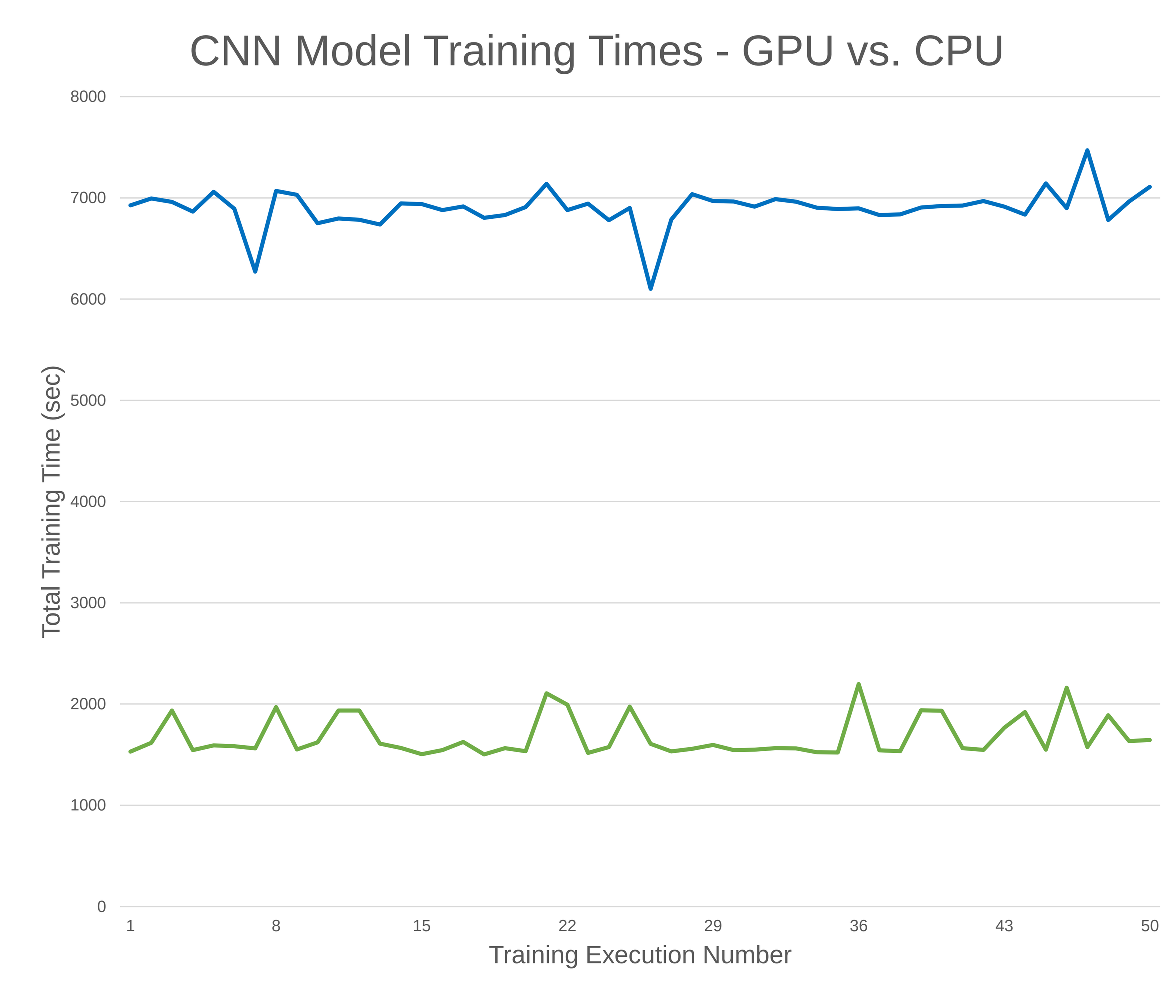
Also using Google Colaboratory for easy GPU and CPU support, I tested my CNN model on images that I collected from a video file. The CPU averaged at about 0.017 seconds per image and GPU averaged at about 0.0019 seconds. While it may not seem like much of a difference in time, adding up hundreds of small operations like these can really make a difference in the long run. The CPU takes about 8.436 times longer than the GPU per image.

2 Tools and Objectives

- NVIDIA TX1 ‘Jet’ Kit
- Arduino Mega
- Bluetooth Support
- Remote control capabilities
- 2 Motors
- Obstacle avoidance
- TX1 communicates serially with Arduino
- GPU vs. CPU Performance



4 Training



With the model that I designed, I ran tests on the time it took to train the model while using GPU and CPU as runtime types. I ran 50 tests for both GPU and CPU, utilizing Google Colaboratory. As you can see, GPU averages at about 1700 seconds, while CPU is around 6900 seconds. The CPU takes about 4.1 times longer than the GPU per training process.

6 Conclusions

There is clear evidence to suggest that compared to using a CPU, using a GPU for the training and inference process of a convolutional neural network accelerates the process tremendously. I was unable to get the model uploaded onto the Jetson robot, so I was unable to determine whether the model was sufficient for obstacle avoidance with the robot. However, with the data collected, it is clear that using a GPU to process the image input from the camera on the robot will greatly increase the efficiency of the model, reducing the inference time marginally. Future work will involve getting the CNN model uploaded on the robot so the accuracy of the model can be assessed, as well as more work in analyzing the effectiveness of using a GPU to accelerate processes.

ApREECE: Applied Research Experience in Electrical and Computer Engineering REU PROGRAM AT OAKLAND UNIVERSITY

This program was supported by the National Science Foundation Grant No EEC-1659650. Additional support was provided by the School of Engineering and Computer Science at Oakland University.