CS 483 Final Project

By: Jake and Nate

1. Goals and Purpose

Our goal is to have a JetBot be able to drive on the road properly while recognizing the U-Turn, left turn, and right turn signs and respond accordingly.

2. Modeling Specification

Our model uses the ResNet-18 model and uses linear regression akin to this course's lab 3. In fact, we did not change any of the code itself from lab 3 and the only difference is the data used for the model.

3. Hardware and track Specification

The JetBot being used has 4 GB of memory, Cortex-A57 CPU, and a 128-core Maxwell GPU. The track consists of blue tape marking the edges of the road and green tape marking the center line of the road.

4. Software Specification

We used the Jupyter notebooks on the JetBot and ssh into the JetBot to do most of our final project. Google Colab was used to train our model and the model was trained using the PyTorch deep learning framework.

5. Process
    - Setup

    The road was designed in a plus shape, and we have printed out each of the three signs. The three signs are then folded and held in place using tape so that it can stand by itself.

    - Data Collection

    We took roughly 150 pictures of the straight paths and about 150 photos of the corners of the roads. We then took around 80 photos of each of the three different signs for a grand total of 563 pictures. Every picture taken has a circle indicating where the JetBot should be going. For the parts where the JetBot needs to go straight we have the circle in the upper part of the photo and when the JetBot needs to turn we have the circle lower in the photo which is closer to the JetBot.

- Data Organization

All our data was in a singular folder that would be used to train the model.

- Training

We used the ResNet-18 model to train our data. We have 90% of our data to be training data and the other 10% the test data. We used the DataLoader class to load the data with a batch size of 8 and allow the data to be shuffled.

- Testing

We did 70 epochs and chose the model that has the smallest loss for the testing data set.

- Running the robot

We ran the JetBot along the road autonomously using our model. We changed some of the parameters such as steering_gain and steering_bias until the JetBot could follow the road and respond to the three signs properly by itself.

6. Critiques

Initially when we started, we used classification to recognize the signs and we did not worry about road following. However, we had to make a new dataset and used regression instead, one of the reasons for this is that taking photos for road following for classification would take more time than restarting from scratch using regression. If we were to do this project again, we would consider how to make the JetBot work for all the required criteria rather than focusing on a subset of that.

For our regression model we would consider getting better training data and paying closer attention to the amount of pictures taken for each scenario that way the model would likely be better trained and not overtrained on anything.

If we were to give recommendations to others, we would highly recommend using regression versus classification for this assignment. We would also suggest learning where to click on the picture telling the JetBot where it should go in relation to itself, as the distance and angle in which you choose will dictate how the bot moves and turns. Lastly, testing is above all the most important thing to do, seeing how the bot acts on its own based on the training set you’ve gathered so far will give you insight on where it needs to actually be improved, so taking smaller data collections at a time and seeing how it performs will make the process a lot easier versus if you try taking all your data in one setting without testing.