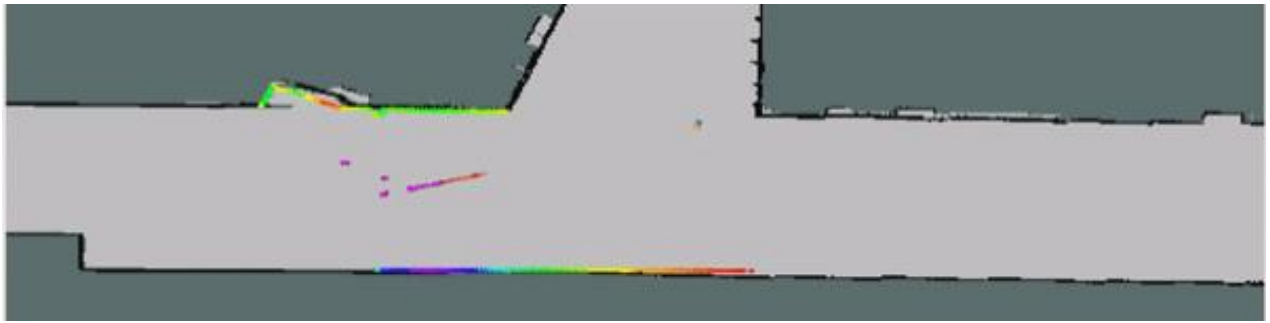
Final Challenge

Team 2:
The Bad Bananas
Isaac, Allie, Gabe,
Vibha, Nathan, Amro

May 15, 2019

BAD BANANA

Using deep learning for vision-only navigation unlocks a new approach to autonomy.

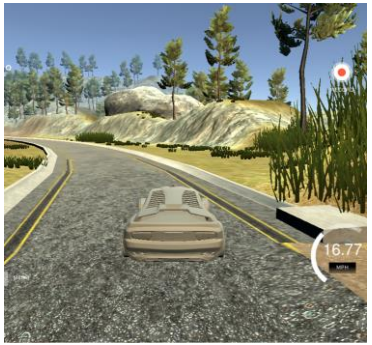


Agenda

Gate Detection and Following

Imitation Learning

Simulation Implementation



Real World Implementation

Gate Detection and Following

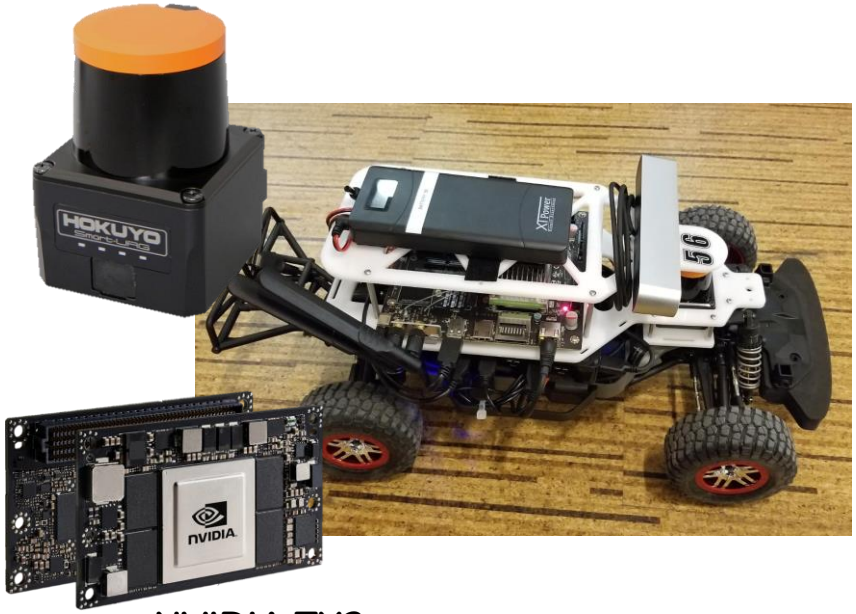
Imitation Learning

Simulation Implementation

Real World Implementation

Before we could start detecting gates, we needed to get used to our new car.

Hokuyo LiDAR

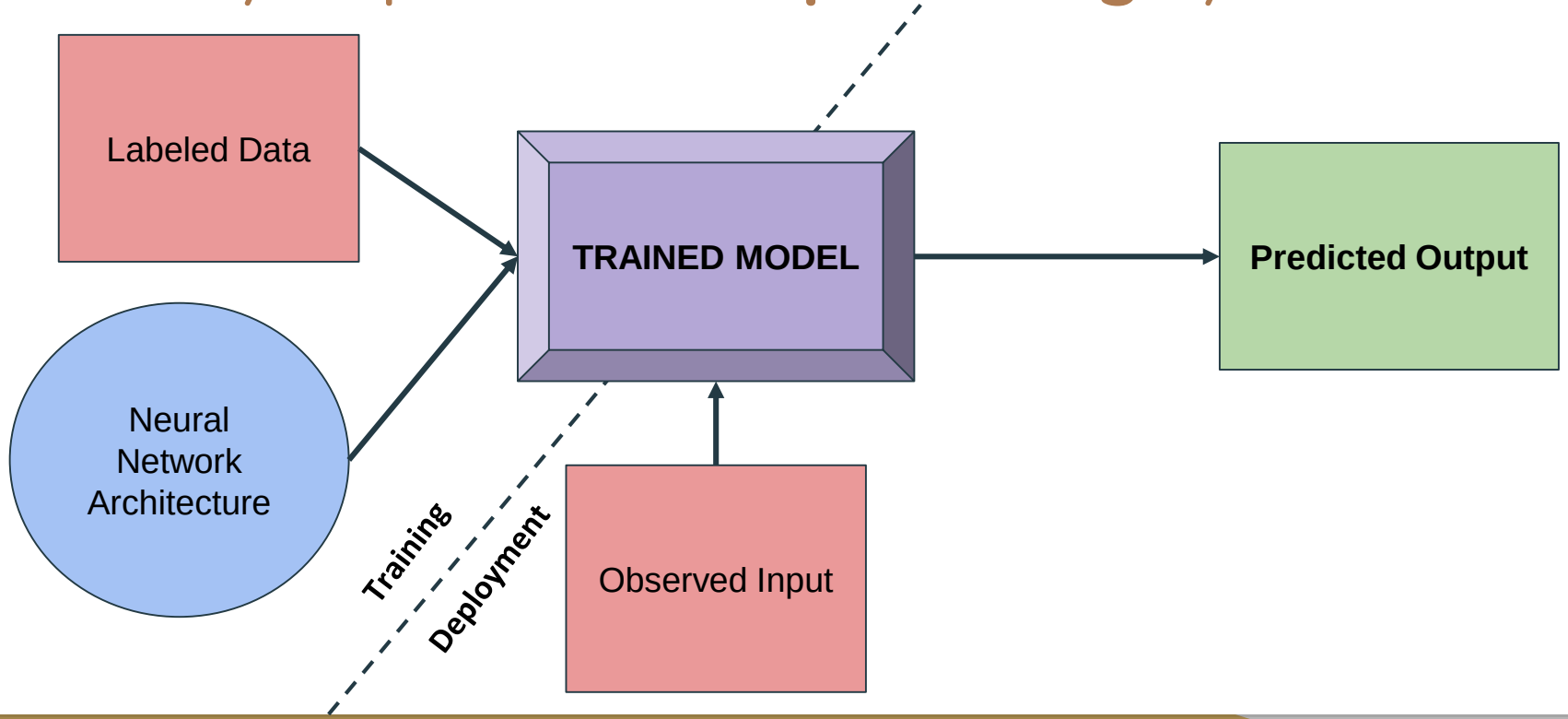


NVIDIA TX2

Logitech Webcam

NVIDIA Xavier

To detect gates with TensorFlow, we created a weakly-supervised **deep learning** system.

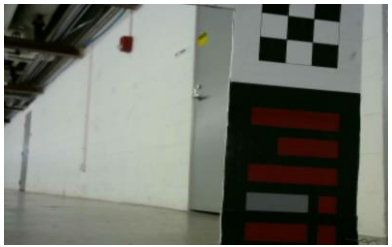


We collected **data** for training and testing our Convolutional Neural Network.

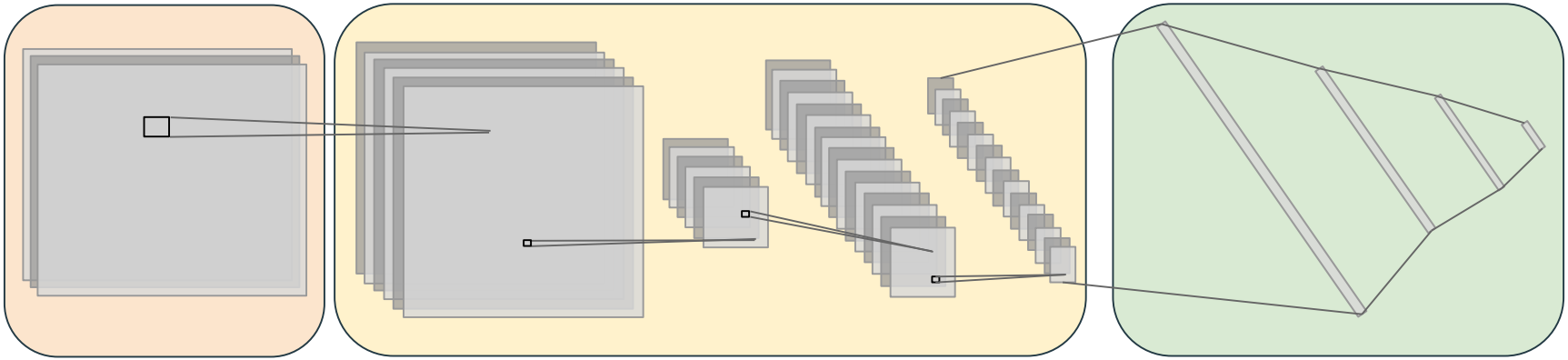
“No Gate”




“Gate”




Our CNN model architecture learns **visual features** with convolutional and pooling layers.

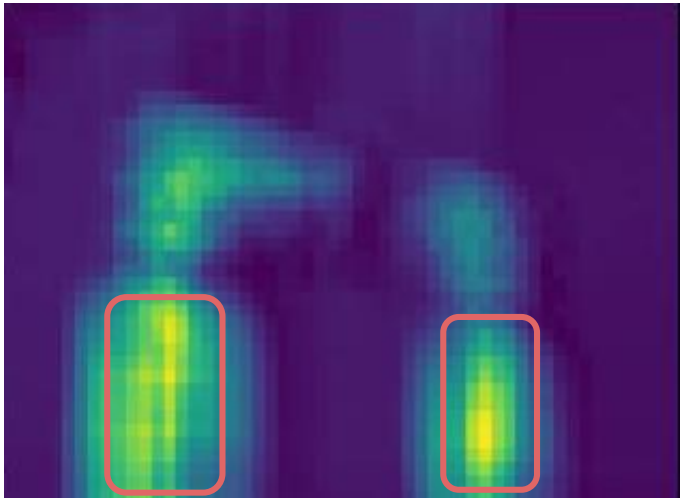


Input Image

Convolutional Layers and Pooling

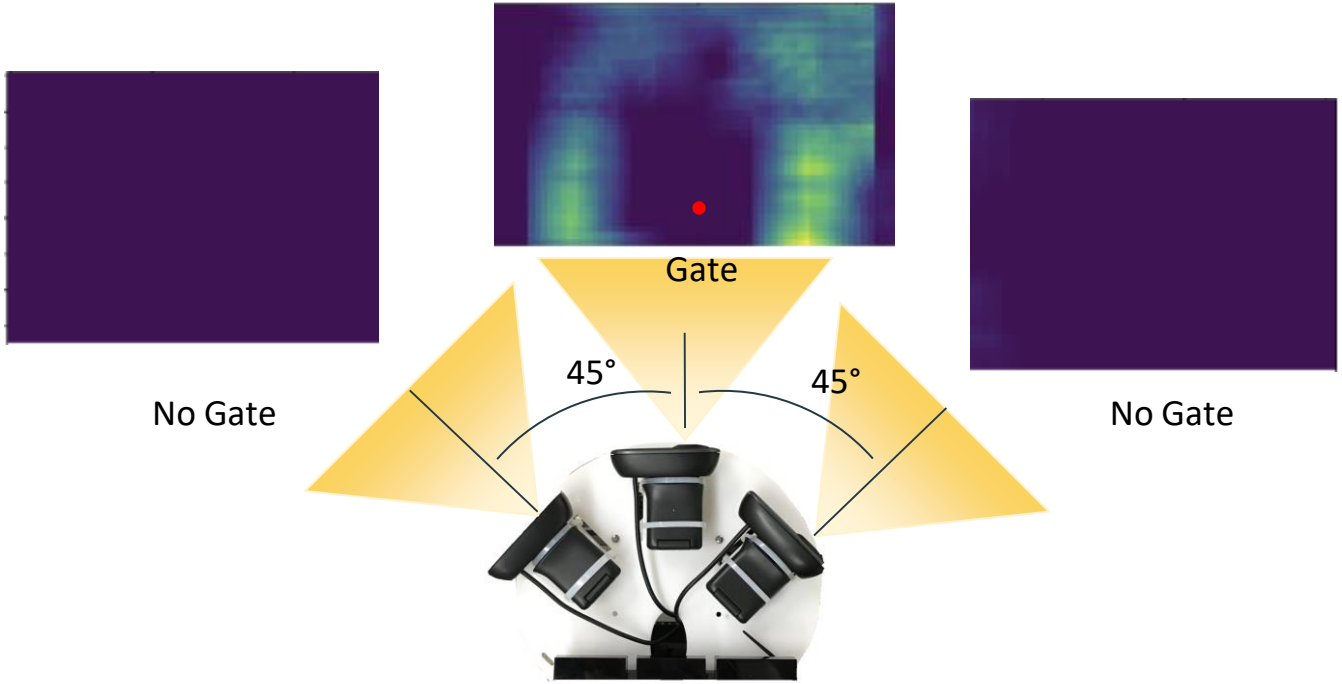
Dense Layers and Classification

The convolution layers in our model generate **heatmaps**, which reveal critical gate features.



Model generally finds black columns.

We use the **heatmap** and the car **camera positions** to compute a steering angle.



Because we are using **weakly-supervised learning**, we don't always make it through gates.

False positives cause a lot of errors.

Low noise environments perform better.

Imitation Learning

Gate Detection and Following

Simulation Implementation

Real World Implementation

Simulation Implementation

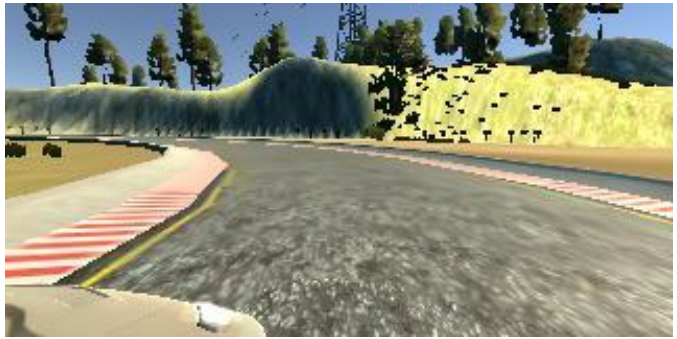
Gate Detection and Following

Imitation Learning

Real World Implementation

We trained a CNN model to implement vision-only navigation with imitation learning.

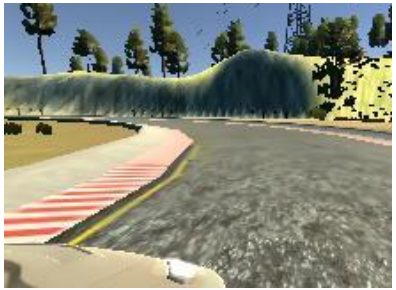
I steer 20° left at this turn.

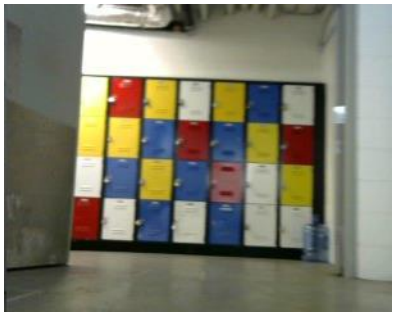


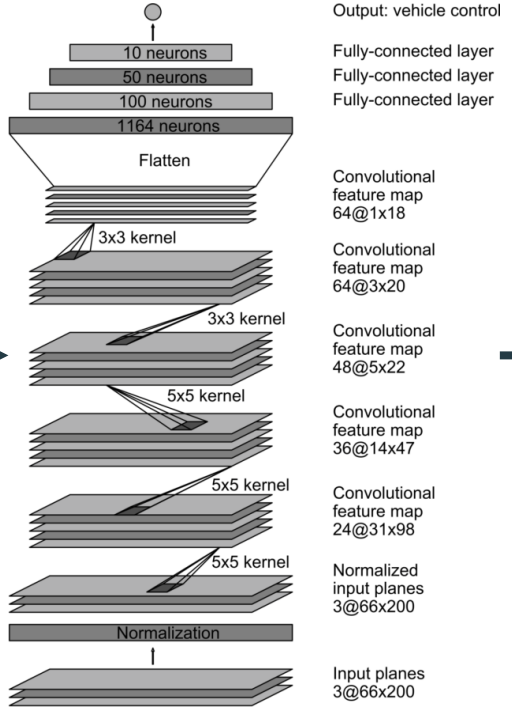
Vehicle Perspective

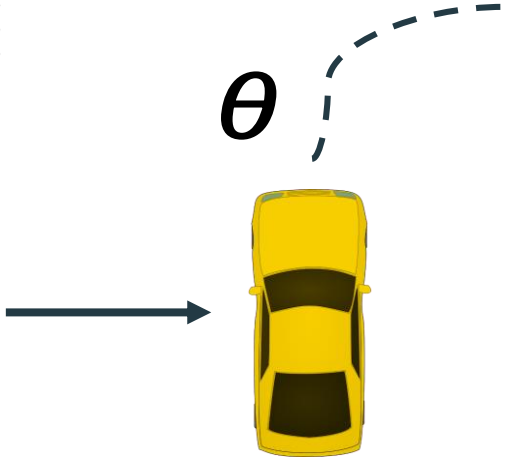
Ok! Next time I see an image like this I will turn 20° left.

We use a CNN model architecture, PilotNet, to output a commanded steering angle.

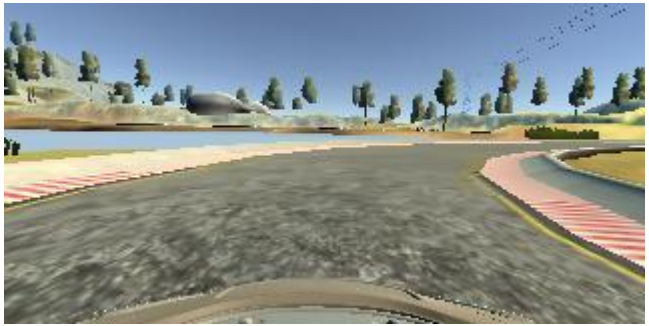








We started by collecting a large amount of data by manually driving in a simulation environment.



Steering angle: 0.45 radians
Speed: 30.15 km/h

To create a more robust data set, we took our data and augmented some of the images

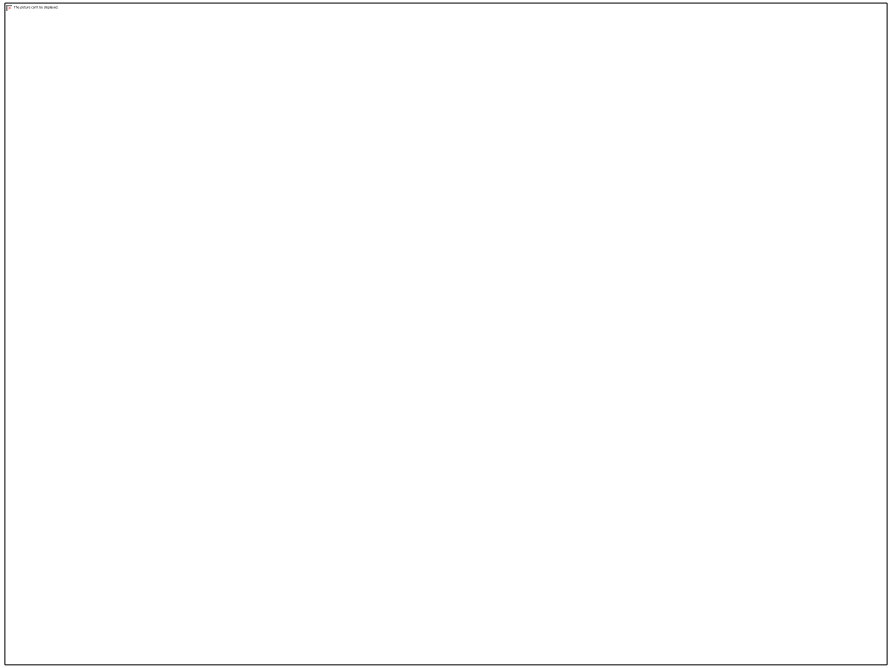
Original

Random Gamma

Random Brightness

Random Translate

We tested our model on the simulator, and the car was able to drive autonomously.



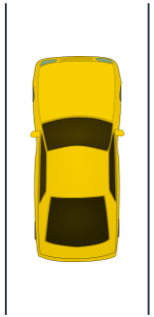
Real World Implementation

Gate Detection and Following

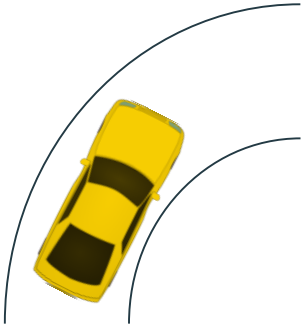
Imitation Learning

Simulation Implementation

After utilizing imitation learning in simulation, we identified needed modifications for the racecar.



VS.

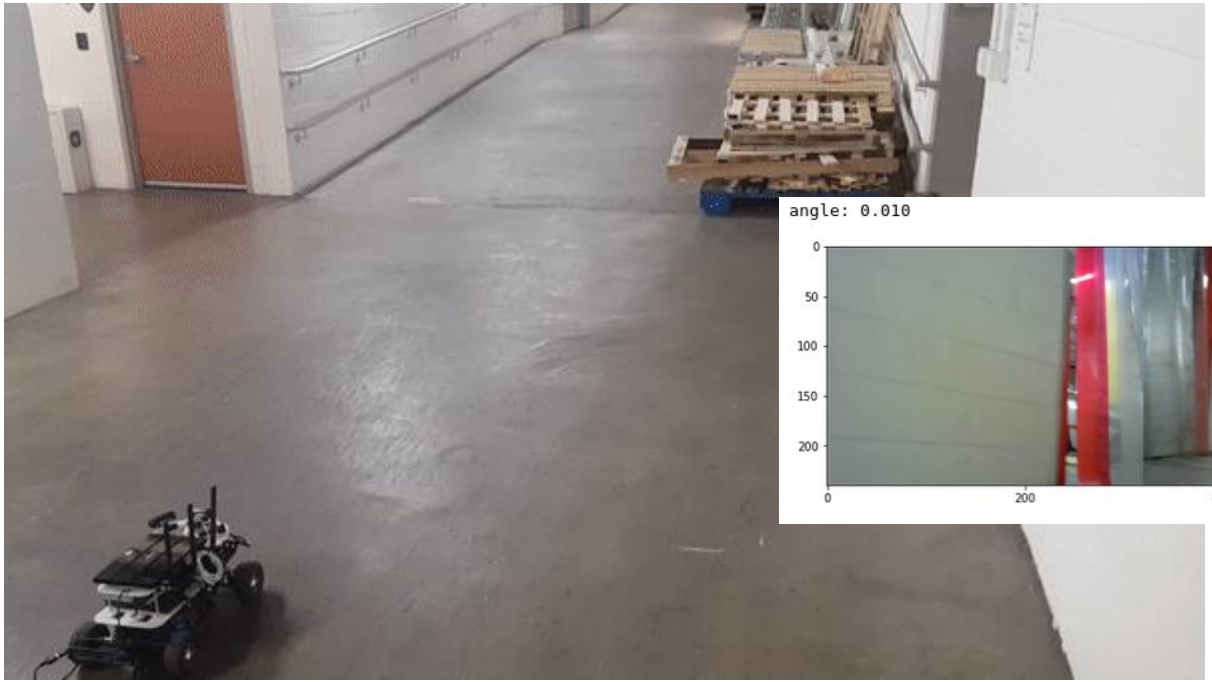


5000 Training Iterations

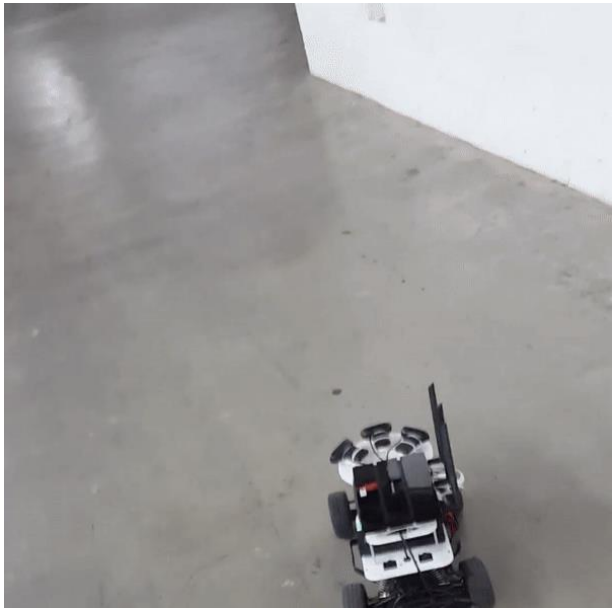
VS.

50000 Training Iterations

Next, we collected multiple laps of manual driving data around the Stata basement loop.



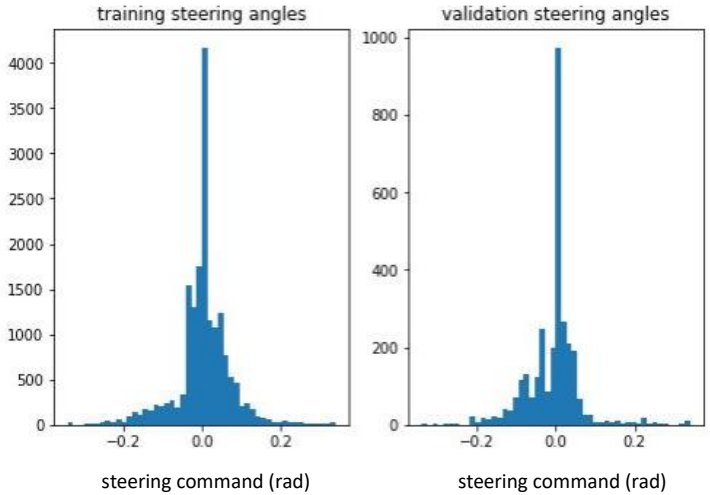
After collecting some training data and training a model, we checked our initial performance…



…we still needed more focused training data and a better approach for creating our models.

To improve the robustness of our Stata loop model, we strategically manipulated our data.

First Attempt

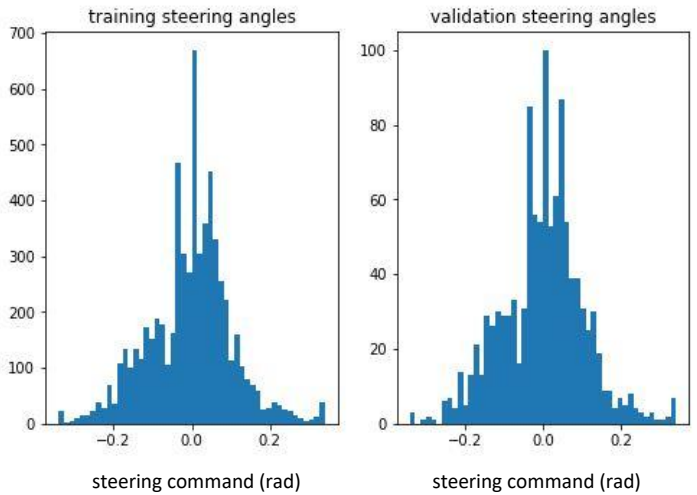


Synthetic Augmentation

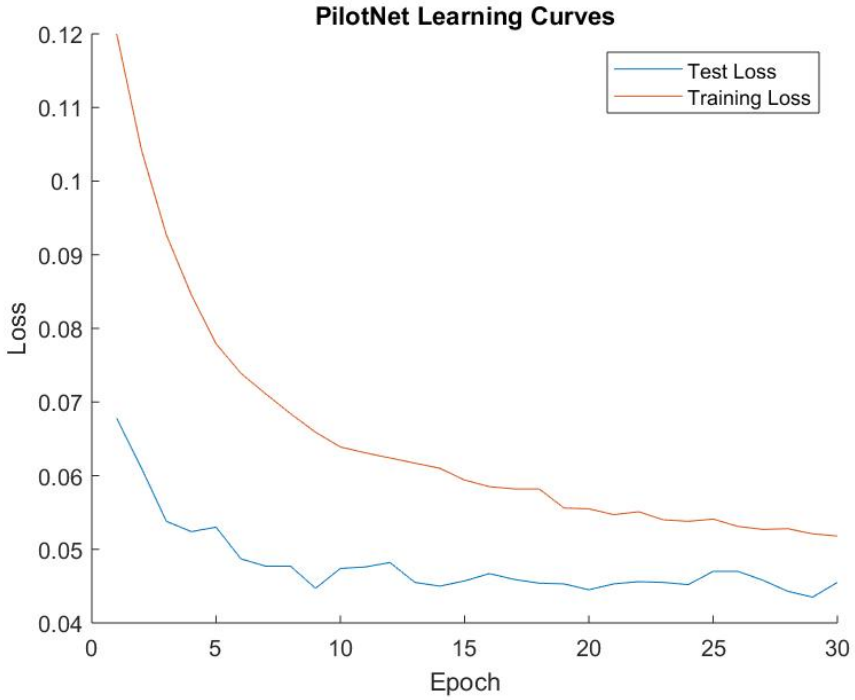
Targeted Data Collection

Targeted Sampling

Final Dataset



We trained PilotNet on a remote server for 60,000 iterations.



Once we had a model with adequate performance, we traversed the Stata loop!

We’re still learning, but we’ve come a long way!

Lessons Learned
Machine learning is difficult to debug and hone.

Team Strengths
Coming together as a team for the final push.

Weaknesses
Time management.

Looking Back
This semester we’ve developed into a strong team!

That's a Wrap!