Team 1 - Lane Detection & Obstacle Avoidance: Final Report

Harvard Summer School

DMGD S-17Table of Contents

# 

# 

[**Overview**](#_73x3ja9w9yrv) **2**

[Main Goals](#_gmqe2m98zgzl) 2

[**Team Members**](#_qsyfxrfdwp0y) **2**

[**Roles**](#_4yk9qj93q29s) **2**

[Role Definitions](#_mry654o1avof) 3

[**Contributions**](#_ubwt4upcexzk) **3**

[**Project Milestones**](#_w1c3sqou3hnp) **5**

[Week By Week Milestones](#_gvvbmi4lar1s) 5

[Weeks 1-3 (06/22/2020 - 07/05/2020)](#_c8ac3ex1o7p) 5

[Week 4 (07/06/2020 - 07/12/2020)](#_w0epn5j6l5oj) 5

[Week 5 (07/13/2020 - 07/19/2020)](#_j6hdcyetxqhg) 5

[Week 6 (07/20/2020 - 07/26/2020)](#_yk0kbds0buon) 6

[Week 7 (07/27/2020 - 08/04/2020)](#_s1wii389ibfu) 6

[Milestone Schedule Performance](#_di6sikf0em65) 6

[**Equipment**](#_1bs8izy4h7lt) **7**

[Hardware](#_et6yvgxrxcgm) 7

[Software & Development Tools](#_wqkgfnbct885) 7

[Development/Collaboration tools](#_qum5h17tcasy) 7

[Software](#_a9qgem8vfkro) 7

[Training Platforms](#_d97zbfa6oeyl) 8

[**Project Summary**](#_riyfozsrwoga) **8**

[**Roadblocks**](#_f21bx0pq2274) **8**

[**Conclusions & Final Thoughts**](#_6bfjx4oxux11) **10**

[Future Endeavors](#_t0m6b34zuknu) 10

# 

# 

# Overview

The creation of autonomous vehicles (AVs) has faced several challenges as mankind has developed up the levels of vehicular automation. Among these major challenges, two that stand out, in particular, are lane detection and obstacle avoidance. Being able to keep an AV inside a lane and aware of obstacles as the road/track around them changes is a major concern that needs to be addressed if a level five (fully autonomous) vehicle is going to be available one day. As such our team endeavored to make various models to combine both lane detection/road following and obstacle/collision avoidance to navigate various tracks.

# 

## Main Goals

As surmised by the name of our team, the main goals of our project were to program our various jetbots to accomplish the tasks of road following/lane detection and collision/obstacle avoidance. In terms of our goal we built individual models for the two aspects of the project and then created a combined demo to demonstrate the capabilities of both models in tandem. These models were created with various tracks of varying shapes and complexities in mind to see if any distinctions between lane detection and obstacle avoidance occurred for our robots.

# Team Members

1. [Khalid Aliweh](mailto:khalidaliweh@gmail.com) - Team Leader/Builder/Programmer
2. [Zane Hernandez](mailto:zah693@g.harvard.edu) - Programmer/Documentation
3. [Ishaan Bhagat](mailto:ishaanbhagat@g.harvard.edu) - Graphic Designer/Videographer/Builder
4. [Alejandro Montero](mailto:alejandromontero@g.harvard.edu) - Programmer/Builder
5. [Favion Harvard](http://fharvard@g.harvard.edu) - Programmer/Builder
6. [Asia Vines](mailto:vines116@teamstudents.org) - Programmer/Documentation
7. [Oscar Anchondo](mailto:oanchondo@g.harvard.edu)- Programmer/Documentation/Builder

# Roles

In our team Roles are defined but they aren’t strict. Due to advice from previous iterations of the class our team chose to incorporate a degree of flexibility in role assignment. While each member was told to select their own role and thus mark their own strengths toward completing the project, each member was expected to branch out into any of the remaining roles. This was done partially to avoid the problems previous teams had faced in making roles too strict from the outset, as well as to allow each member of the team to try their hand at each and every possible task and get the most out of this class. Some basic descriptions for each role can be found below this paragraph.

## Role Definitions

* Programmer (will work on developing the code and training the models)
* Builder (will build the tracks and test the bot on them)
* Documentation (will compile reports and presentation documents)
* Graphic Designer/Videographer (Editing for the final project slides and videos)
* Leadership (Communication and presentation of the project)

# Contributions

Khalid Aliweh - I proposed the project Idea and helped kick off the project by setting up the Github repo, Trello board, and connected it to TeamGantt. Worked with the team on refining the project plan and collaborated on creating the project proposal and defining the team roles. Created the required Trello board tasks and subtasks to complete the project. Built my bot and a two-lane track. Collected 3000 images for the road following model and trained 17 different models for the task. Experimented with different image augmentation techniques, using pytorch ignite to measure the model accuracy and experimented with training the models from scratch vs using transfer learning. Collected ~200 images for the obstacle avoidance models and trained the model. I created the final notebook to combine the obstacle avoidance and road following models. Attended all the group meetings and contributed to the creation of the final presentation. Researched solutions to enhance the lane following model and bot driving, shared my findings with the team and helped with solving some of the technical issues. In addition, I represented the team in the lectures and the final presentation.

Zane Hernandez - My main contributions to the team project was organization and creation of team documents. I set up the presentation slides and final project report to be filled out by all members of the team while also contributing to many of the sections for both documents. In addition I generally served as the one to check for clarification of project requirements from professor Ramirez and the teaching assistants (e.g. checking that we each had to submit a 2-3 minute video of our project demo). I attended every team meeting and generally took meeting notes for lingering questions about the project. I also took it upon myself to assist Asia in completing her requirements for submitting the tutorials. As for actual components of the demonstration of the project, I built a simple ovular track with only one lane to compare and contrast with my teammates tracks that generally had at least two lanes and more complex shapes to see if any interesting differences in our models appeared. I provided over one hundred photos for my own obstacle/collision avoidance model and over four hundred photos for my lane detection/road following model in various natural and artificial lighting conditions. I achieved some success in combining the two models. My jetbot avoids collisions expertly, but the lane detection is a bit rough.

Ishaan Bhagat - My main role was as the team’s head of communication and collaboration. I set up and supervised our group’s main form of communication, our Slack channel, while remaining active on it as much as I could. I also coordinated and set up our group’s biweekly virtual ‘check-ins’ or meetings. As for contributing to our main objective of the project, I acted as a builder and programmer, constructing a single-lane ‘P-shape’ track, to really test the Jetbot’s capability to handle a large amount of precise turns. I provided over 600 photos for my lane following model, and over 400 photos for my obstacle avoidance model. At the end, I did manage to successfully achieve our group’s objective of combining lane following with obstacle avoidance to a high level of precision.

Alejandro Montero - I helped set up some tools for team organization and work submission, like the YouTube channel. I also provided casual help to team efforts whenever it was required from me. My main role was to help as a builder and programmer. I built the Jetbot, and also set up a 2 m2 oval shaped track with two lanes to test it on. I gathered over 200 photos for my personal collision avoidance model, and 250 photos for my lane following model. I was able to successfully run the collision avoidance model smoothly, but admittedly encountered some troubles with the lane following model.

Favion Harvard - With the project, I really felt as though I did not help a lot because of me by being as knowledgeable as the other teammates. I tried to contribute with the zoom meeting when I could, and tried my best to figure out all the software that was being used in the project. I feel as though I would have done better actually being with the group in person, and being able to ask questions and get help more easily. I wish that I could have done more for the group then just building my robot, and trying to program, but I tried.

Asia Vines - Due to a late shipment for my jetbot and the lack of necessities, I wasn’t eligible enough to work on the physical setup for the Jetbot, but I concluded with writing out descriptions upon the faced challenges as well as the segments of code that played part for each tutorial, specifically from the Jupyter notebooks and the Github repository. I was close enough to working with a simulator, since I was able to generate a basic background along with obstacles, but, due to numerous errors and the need of certain features and packages, I had no choice but to resign. However, the understanding of the tutorials and confronted issues gave me a boost into successfully working on my jetbot for the remainder of the summer. At least, by then, I’ll have all the systematic and physical components that I need.

Oscar Anchondo - Setup and coordinated the inception of our meeting schedule per member’s availability. Made contributions, edits, and formatting to the team’s documentation including project proposal, presentation, final report, and Github repository. Supplied miscellaneous ad-hoc assistance to team efforts and provided anonymous support on hardware setup, operation, and troubleshooting to classmates via Piazza. Collected nearly 1500 labeled images for obstacle avoidance and close to 300 for lane following. Successfully implemented the combined model around a wide lane oval shaped track in a busy and compact environment.

# Project Milestones

## 

## Week By Week Milestones

The following milestones for the project were planned back at the beginning of the course when we were preparing the project proposal.

### Weeks 1-3 (06/22/2020 - 07/05/2020)

* Purchasing materials
* Building robot
* Setting up the environments
  + Github
  + Slack
  + Trello
  + Google Doc/Drive

### Week 4 (07/06/2020 - 07/12/2020)

* Build tracks
* Collect data from track
* Collect data for the object avoidance model

### Week 5 (07/13/2020 - 07/19/2020)

* Train the road following model
* Run the road following demo
* Retrain the models as needed
* Train the custom object avoidance model
* Run the customized object avoidance demo
* Retrain the models as needed
* Combine the road following and object avoidance models
* Start testing

### Week 6 (07/20/2020 - 07/26/2020)

* Test - and retrain as needed.
* Assemble Videos for the presentation

### Week 7 (07/27/2020 - 08/04/2020)

* Finalize Project
* Create the Project Report
* Presentation

### 

## Milestone Schedule Performance

For the first few four weeks of our project our team stayed on track with the timeline of events detailed above. However, due to various issues, ranging from issues receiving materials, due to the current world situation, to struggles with affixing the resnet and alexnet models we started to fall behind schedule starting in week five with only roughly thirty-three percent of the project being completed. We were able to bounce back rather quickly once our lagging team members managed to acquire what was needed to complete the project, we each managed to pick up the pace and make up for lost time and knocking out up sixty-six percent at the end of week six. We were able to finalize the project toward the end of week seven which while not ideal went off with only a few minor hitches in cleaning up our repository. Overall we’d say our experience is a good example of persevering through adversity.

# Equipment

## Hardware

Our team had various background hardware in the development of this project. To complete the combined lane detection & obstacle avoidance model we made use of the following hardware to run the courses:

* [Nvidia Jetson Nano Developer Kit](https://developer.nvidia.com/embedded/jetson-nano-developer-kit)
* [Waveshare JetBot AI Kit](https://www.waveshare.com/wiki/JetBot_AI_Kit)
* [Sparkfun JetBot kit](https://www.sparkfun.com/Jetson)
* Non-specific Robot [Camera 2, WiFi 1, Wheels 2] (Built from [Bill of Materials](https://github.com/NVIDIA-AI-IOT/jetbot/wiki/Bill-of-Materials))

## Software & Development Tools

Our team had various software and tools in the development of this project. To complete the combined lane detection & obstacle avoidance model we made use of the following software to program the our jetbots and coordinate our team:

### Development/Collaboration tools

* [GitHub Repo](https://code.harvard.edu/kha000/ldoa)
* [Slack](https://app.slack.com/client/T0176DFS996/C016GJ9PNQJ)
* [Trello](https://trello.com/b/Nu754K2r/lane-detection-and-object-avoidance)
* [YouTube Channel](https://www.youtube.com/channel/UCJst-iHjQbA3Y4g8UToJQGQ)
* [Proposal](https://docs.google.com/document/d/196blFT1lBQ1a5TE2bOorxXmHZZm_3A8WiqmIQTvXhOA/edit#heading=h.qmhkdk35500b)
* [Presentation](https://docs.google.com/presentation/d/1QLed1v8lgorhKyQP95bi0CgoM4ivz4DWdWcMuZBAdGo/edit?usp=sharing)
* [Team GANTT](https://prod.teamgantt.com/gantt/schedule/?ids=2255786#ids=2255786&user=&custom=&company=&hide_completed=false&date_filter=&color_filter=)
* Jetbot JupyterLabs
* Zoom

### Software

* Python Software Language
* Pytorch
* CV2
* Resnet18, alexnet Models
* Numpy
* Pillow
* Jetbot prebuilt python drivers

### Training Platforms

* Khalid ‘s Machine (OS, GPU)
* Google Colab (GPU)
* Jetbot- Nvidia Jetson Nano(s) ([Technical Specifications](https://developer.nvidia.com/embedded/jetson-nano-developer-kit))

# Project Summary

The final demo combines pre-trained convolutional neural networks (CNNs) fine tuned using each of the team members’ data to better suit our navigation and obstacle avoidance ambitions. To navigate around the track, Torchvision’s pretrained [Alexnet](https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py) were fine tuned with over 250 images labeled with a target X, Y values used to compute steering values. As for detecting obstacles, [ResNet-18](https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py) was fine tuned with datasets containing as many as 1500 labeled images in a variety of environments and scenarios.

Once the transfer learning process is done, both models are loaded into the demo notebook environment where a pre-processing function is defined to convert the format of images from that of our cameras to one suitable for our neural networks. This is done by leveraging Torch, CV2, Pillow, and NumPy libraries that aid in converting BRG to RGB, HWC to CHW layout and normalize training parameters. Once the function does the conversion, then it directs all data into the GPU and initializes a batch dimension. Before the pre-processing function is called the robot, camera, and all the control and information sliders as well their respective connector functions are initialized and defined such that the robot’s view is displayed before the robot moves. Upon setup, then the function that deals with changing views as the robot moves is defined and executed. This same function will then call the preprocessing function every time the robot sees something new, then pass the output to the respective neural network , calculate steering, control individual motor thrust, compute obstacle probability, and manage potential action. Eventually, the execution of the combined model allows the robots to successfully steer through a variety of tracks depending on the fine tuning data used during the transfer learning process, accurately detect obstacles, and stop accordingly. Once obstacles are removed from the track and no longer detected, then the robot resumes motion. Video examples of this process can be reached through the Github repository listed above along with our presentation slides.

# Roadblocks

As one likely expects, this project did not come together without a few hardships along the way. Some were simple and easily resolved, others were more complex and took time to overcome, and others still were beyond our control. First and foremost among these roadblocks was the structure of the project team. Unlike previous iterations of the class we are a fully remote team separated by great distances. While a good portion of us were on the east coast of the United States, others were located around the world placing us in very different time zones. This greatly affected our means of communication compared to local teams of previous classes. This fully remote setup was due to the ongoing coronavirus pandemic which itself has thrown numerous obstacles in this class’s path. One such obstacle felt by at least some of our team members during this pandemic was a delay in acquiring materials to build and set up our robots and thus hampered our ability to perform the project. Given these issues we quickly established connection with one another and set up meeting schedules and in-team status updates while we coped with the current situation.

In addition to the remote team aspect, our various members had to deal with quite the learning curve. While some of us had experience in using our development tools (such as GitHub for team project repositories) and commands (e.g. scp to upload and download to and from our robots) others had never made use of them in these scenarios, thus our more experienced members had to catch up the more inexperienced ones. Beyond that some of our team members also had to deal with our own individual poor WiFi connectivity and local power outages from inclement weather hindering our abilities to work as efficiently as possible. We ultimately worked around these issues by finding alternate WiFi networks, in the case of the former, and waiting out outages in the case of inclement weather. There were also a few issues with processing captured recordings of demos for at least one of our member’s in the form of corrupted video files. This was gotten around through small bits of repair and editing performed by software such as iMovie, as well as through Google troubleshooting when possible.

Another aspect of issues we faced in performing our project came from the robots and project code themselves. One such example came from one of our members robot’s inexplicably changing its IP address multiple times throughout the course of the project. While minor, it did produce some headaches in having to reinitialize JupyterLab and update the browser and secure shell to match the new address. Other such issues included delays in our JupyterLab environments due to code specifications such as streaming to the browser causing massive lag or server disconnection. This was however easily handled through careful selection of camera fps, activation of the jetbot fans, and removing the code to display the stream of jetbot webcams from the lab notebooks, though it did limit some members collecting extra data in the end. There were also issues with data collection. One particular note was the use of gamepad controllers to collect the angle points of our road following models. Using the analog sticks of said controllers proved fairly finicky with rapid shifts of projected route for the slightest shift in stick position thus making it difficult to capture our routes from point to point. To counteract this, several of our members resorted to using the browser sliders to get a more precise flow around our tracks. Furthermore, during the lane following portions of the demo, there were instances across multiple team members where the robot wobbled from side to side when moving in straight lines. To minimize this, the steering gain and thrust speed settings were adjusted to each individual track.

# Conclusions & Final Thoughts

From our work on this project we ended up drawing a few conclusions and pieces of advice that we would like to share for future iterations of this class and their teams. First and foremost you and your teammates must be coordinated. Get into contact with each other as soon as possible and get to drafting how your team will function squared away right from the start with things like github, trello, slack, zoom, etc. Next, when working on your projects be it from building hardware, programming your robot, or any miscellaneous task patience and persistence are virtues that you need to live by. No one solution is necessarily going to work the first time, so keep your head up and be prepared to keep trying. Thirdly make sure that you check out the various wikis connected to your type of jetbot. Oftentimes a quick search through their pages can save you hours of head scratching and internet searches to solve specific problems. Fourthly remember that you and your teammates are not alone in making your project a success. You are part of a class filled with people (both students and the teaching staff) ready and willing to help and support you as you develop your project and work toward completing your goals of this course. Lastly as a tip for any project making your own tracks for autonomous driving aspects, we recommend that you acquire more materials than you need. As with being persistent in the face of difficulty you’re likely going to have to make adjustments so it can only help to have additional materials on hand.

## Future Endeavors

In the event that we were able to continue the project beyond the timeframe of this course, we have a few ideas on how the project would expand. First and foremost is adding additional data to our basic road following/lane detection model to allow the jetbot to perform to an even higher degree of excellence. This would be further expanded upon by adding data from all of the tracks in many different environments (e.g. alternate lighting conditions, no visible lanes, etc.) to better enrich the bots response as it observes the track. We would also plan to combine the various models created by our members to build a new widespread model involving all of the provided data from each track. Secondly we would attach encoders to our jetbots and enhance the current PID controller to allow for smoother riding around our tracks. Finally we would also build onto our current obstacle avoidance model. In its current state the model only detects two classes: blocked and free. With more time we would expand upon the potential datasets to include other classes such as determining lane availability and road sign detection to cover a wider range of situations that are normally seen on real roads.