Autonomous Driving: Path Generation and Route Following with Image and Command Inputs

Caitlin Johnson, PI: Professor Eugene Vorobeychik
Department of Computer Science, Washington University in St. Louis, MO

Introduction

Machine learning is crucial for self-driving cars, and often includes image-based and numerical command inputs to develop autonomous control and decision-making. The central areas of investigation explored in this project are:

1. **High level planning utilizing command inputs.**
2. **Local path generation with lane following and intersection navigation.**
3. **Reinforcement learning with image and command inputs.**

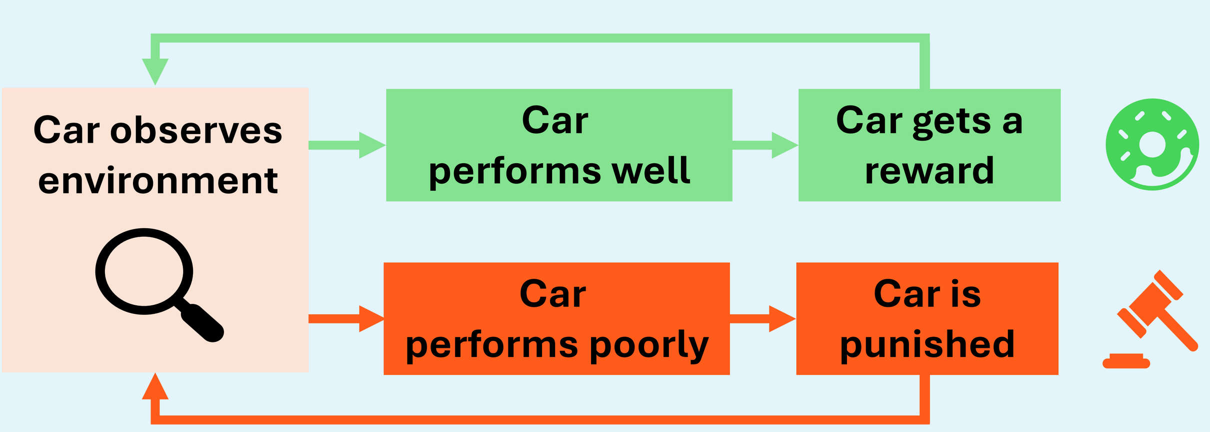
Method Overview

This approach to further developing self-driving vehicles involves a combination of machine learning techniques to enable them to execute commands while maintaining awareness of their surroundings:

- **Convolutional Neural Networks (CNNs):** identifying features in a camera image to generate local paths for the car.

Figure 1: Path annotation for CNN training.

- **Reinforcement learning (RL)**: punishing or rewarding the car when it takes actions to encourage desired behavior.



Step 1: High Level Planning

Self-driving cars usually have a specific destination, and are trained to get there as soon as possible while maintaining safety.

- Our route implementation involves determining the shortest path to a goal location, specifying when to turn right, left, or go straight through an intersection, and specifying when to lane follow:

[(start location, command input), (2nd location, command input), ... , (goal location, command input)]

- The **locations** are represented by (x, y, z) coordinates, and the **command inputs** are represented by integers 0-2 (0: turn left, 1: lane follow, 2: turn right).
- As the car moves along this route, it receives a new command once it reaches the next location in the list.

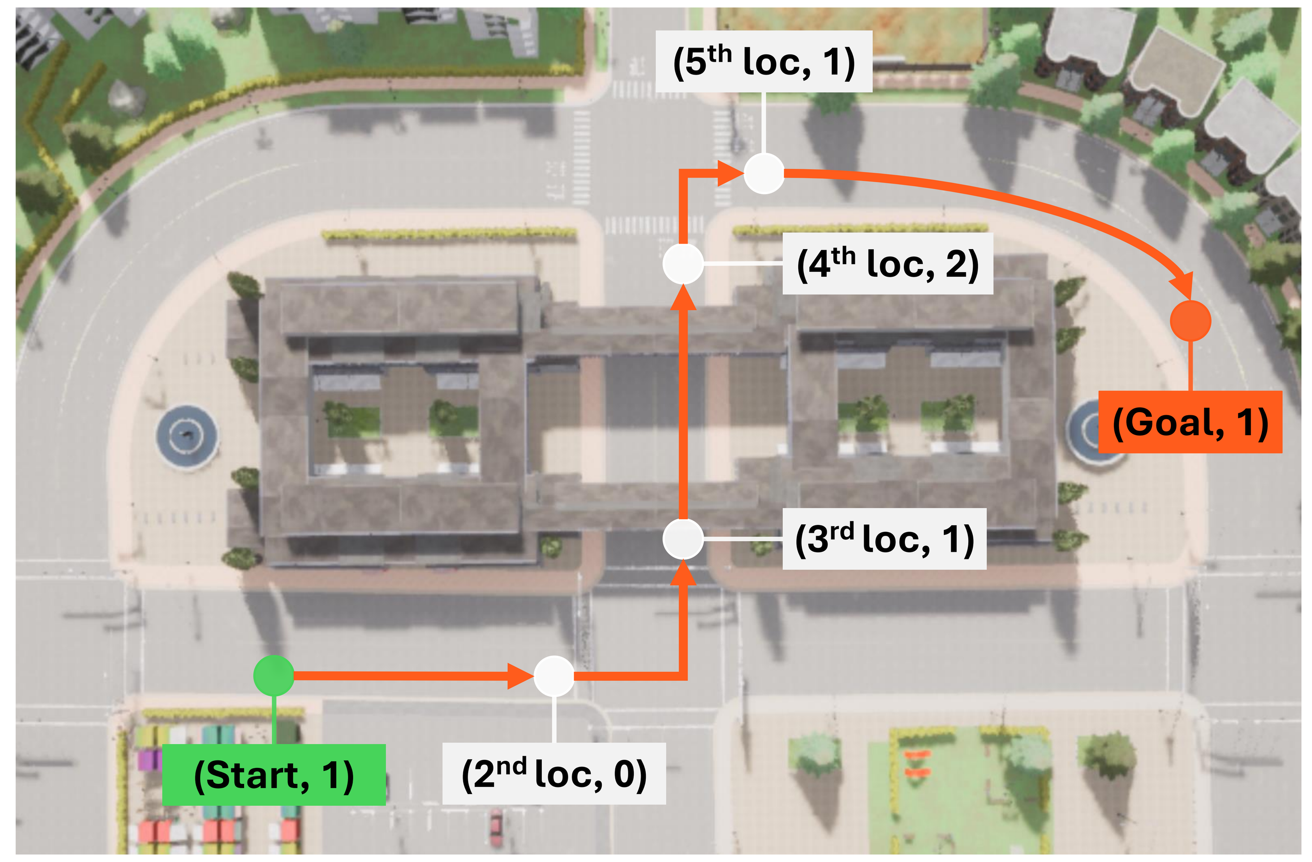


Figure 2: Example route planning, from start to goal.

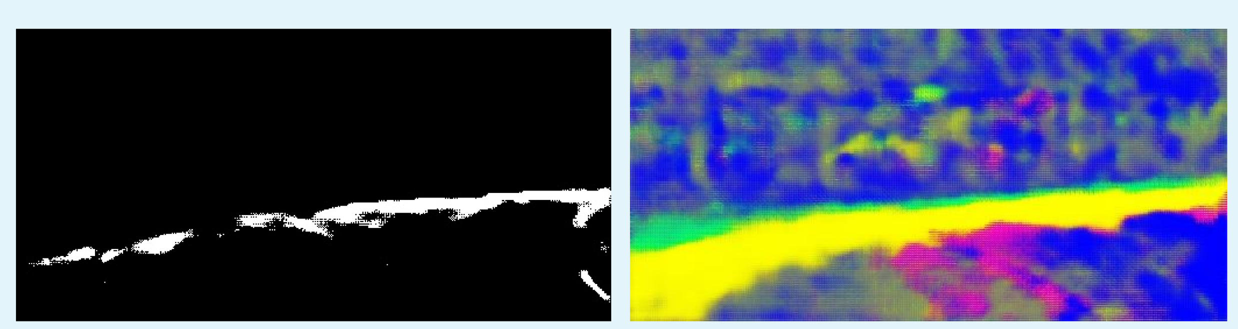
Step 2: Local Path Generation

The car can follow a general path once the high-level plan is created, but the car still needs to navigate locally. LaneNet is a CNN architecture that can be trained to generate local paths.[1]

- The car takes a picture of its surroundings at each step:

Figure 3: Example right turn image input for LaneNet training.

- We can give the car a sense of which sections to stay in-between to facilitate lane following and turning at intersections, even if there are no visible lane markings:



Figures 4, 5: Example LaneNet binary and instance outputs for right turns.

By training a model to convert complex RGB images to binary images, the local path is easier to detect and follow.

Step 3: Navigating Lanes & Intersections

Now the car knows if it should turn or lane follow and it can understand its camera inputs, but it still must learn how to drive.

- In RL models for autonomous driving, cars are often rewarded on aspects like safety, comfort, speed, displacement, traffic rules, etc.[2]
- This project experimented with the following reward function:

$$R_{step} = l + s + c$$

- l: the car's **lateral distance** from the center of its lane.
- s: how close the car's **steering angle** is to the target angle.
- c: -1 if a **collision** occurs, 0 otherwise.

Results

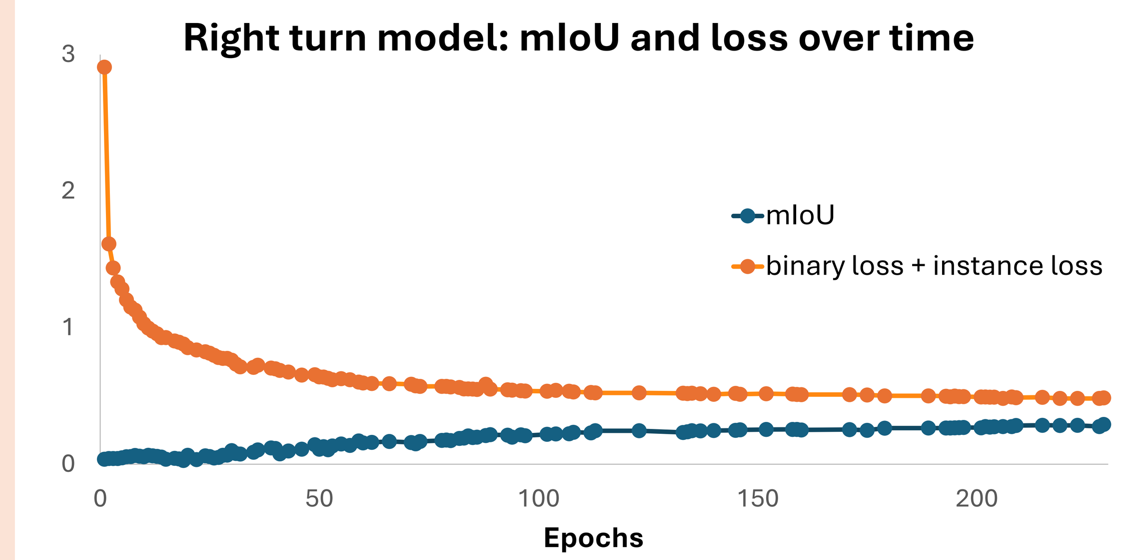


Figure 6: mIoU and loss results for generating right turn paths.

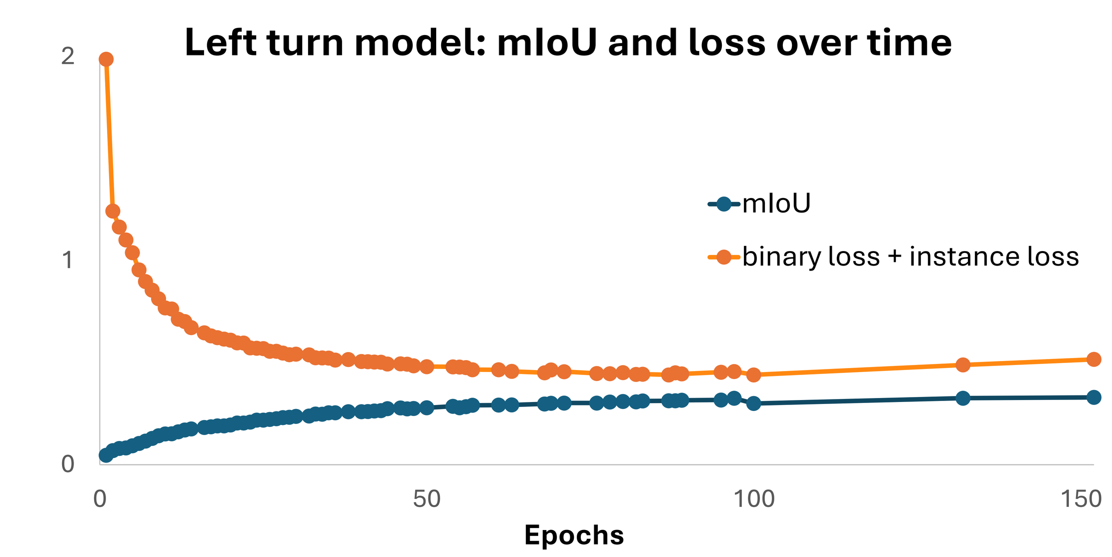


Figure 7: mIoU and loss results for generating left turn paths.

Path generation results: Mean intersection over union (**mIoU**) represents the pixel overlap between the model's generated lane regions and the actual regions, and **loss** represents the model's overall prediction error.

- The right and left turn models reached mIoU = .29, .33 and loss = .47, .51 respectively.
- The lane following model was already trained with mIoU = .57 and loss = .12.

Figures 8, 9: Lane follow command before and after training.
Figures 10, 11: Right turn command before and after training.
Figures 12, 13: Left turn command before and after training.

Before and after training examples of the car receiving commands:

- **Lane follow:** the untrained model veers right; the trained model can follow straight lanes and go through intersections, but does not handle curved lanes well.
- **Right turn:** the untrained model veers right; the trained model steers into the rightmost lane at the end of the intersection.
- **Left turn:** the untrained model veers right; the trained model steers left at first, but then goes straight.

Discussion & Future Work

Local path generation: path generation in areas with curved or nonexistent lane markings proves challenging. Additional model inputs, such as contextual inputs like image history, are worth investigating to further improve results. These inputs could provide the model with a better understanding of the spatial and temporal aspects of the road and vehicle trajectory.

Reinforcement learning with command inputs: the results indicate a slight improvement in lane following and intersection navigation. Using only vision has inherent limitations, potentially manifested by the car's reluctance to steer right or left. Additional sensors (LiDAR, GPS, IMU, etc.) and a modified model and reward function, are worth studying to further improve perception and stability. Overall, this research underscores the importance of multimodal approaches for reliable autonomous driving.

References & Acknowledgements

Special thanks to Owen Ma, George Gao, David Brodsky, Angelo Benoit, and Professor Eugene Vorobeychik for their previous work on lane detection and following, and for their continued guidance and support during this project. This work was supported by the National Science Foundation's (NSF) Research Experience for Undergraduates (REU) program.

1. D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans and L. V. Gool, "Towards End-to-End Lane Detection: an Instance Segmentation Approach," *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 286-291, doi: 10.1109/IVS.2018.8500547.
2. A. Abouelazm, J. Michel and J. M. Zöllner, "A Review of Reward Functions for Reinforcement Learning in the context of Autonomous Driving," *2024 IEEE Intelligent Vehicles Symposium (IV)*, 2024, pp. 156-163, doi: 10.1109/IV55156.2024.10588385.