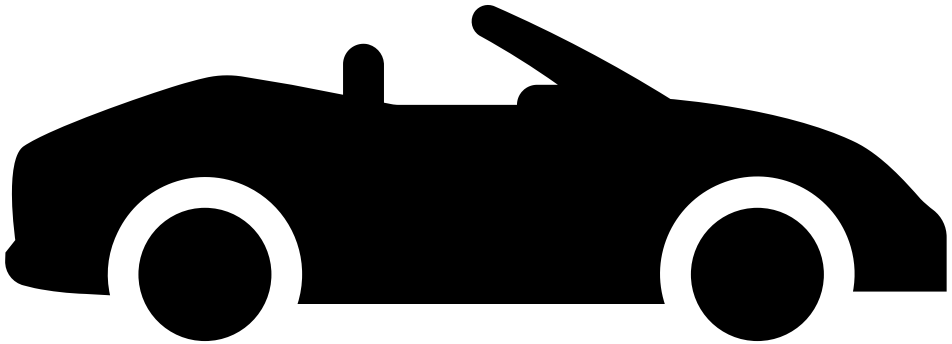


END-TO-END SELF-DRIVING SIMULATION

CHENHAO FANG

HAO JIANG

KOU WANG

TABLE OF CONTENTS

INTRODUCTION

AUTONOMOUS DRIVING TECHNOLOGY

- Autonomous driving technology allows vehicles to self-drive
- Background: a lot of people die in traffic accidents, and traffics are getting worse due to the increasement of the vehicles in the world
- Offers a solution to traffic problem
- Core technology: deep learning

SELF-DRIVING CAR (SDC) IN SIMULATOR

- We implemented a self-driving car in simulator
- Udacity self-driving Car Engineer
- Mannual mode & Autonomous mode
- In the end of the project, our self-driving car (SDC) is able to drive autonomously in the designed track, which contains different road and background conditions

INPUT OF SELF-DRIVING CAR

- In the simulator, the car is able to collect image data from three different cameras: left, middle, and right
- Images are gathered and augmented, transformed into a different to achieve better model performance
- These images are used to train our model

CAMERA VIEWS

Left Camera

Center Camera

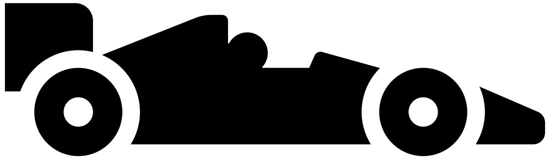
Right Camera

CONVOLUTIONAL NEURAL NETWORK MODEL

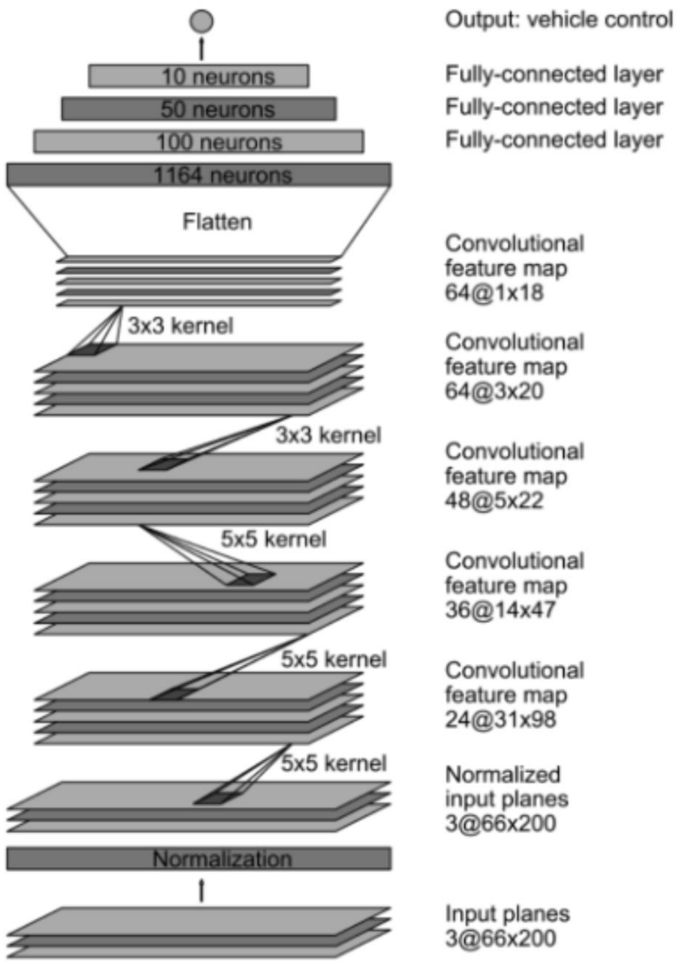
- Trained several CNN model for our self-driving car
- Observed their robustness and accuracy
- Tuned a lot of parameters and hyperparameters
- The final one which contains 1 normalization layer, 5 convolution layer, 3 fully-connected layer and 1 drop-out layer having best model performance

EVALUATION

- Human intervention gets involved when we think it is necessary
- We count the autonomous time and human intervention time
- The ratio of human intervention time and the autonomous time is used for evaluating the model performance: The less the ratio is, the better the model performance is



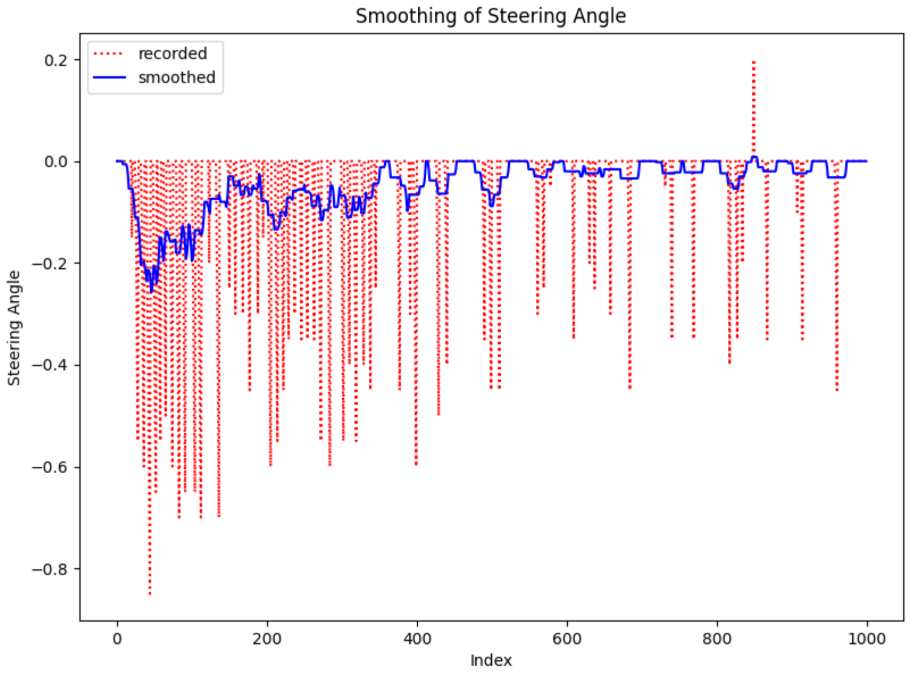
MODEL DESIGN



MODEL ARCHITECTURE

TRAINING AND RESULTS

TRAINING – STEERING ANGLE SMOOTHING



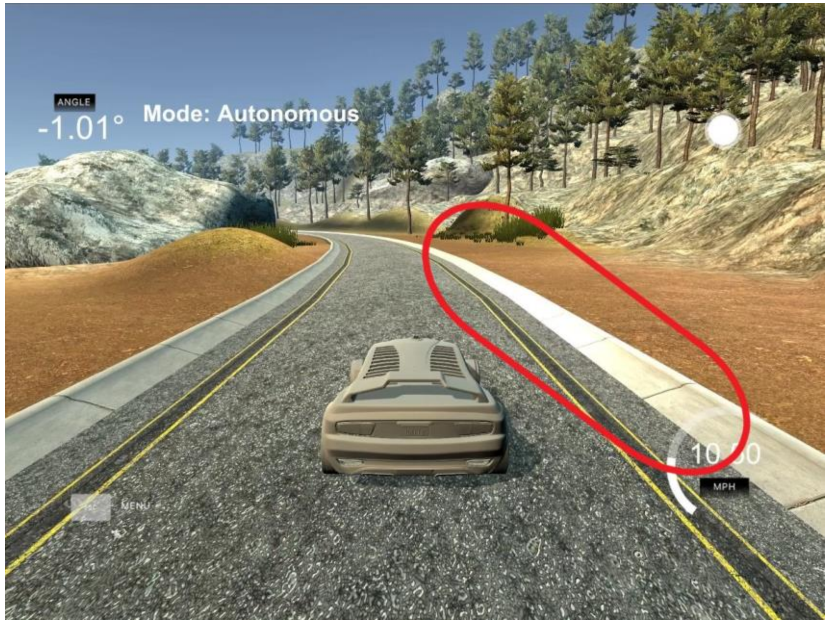
- Due to the problem of the simulator, the recored steering angle can fluctuate a lot
- Difficult to minimize loss function
- Smoothing is used to solve the problem

TRAINING – OVERFITTING

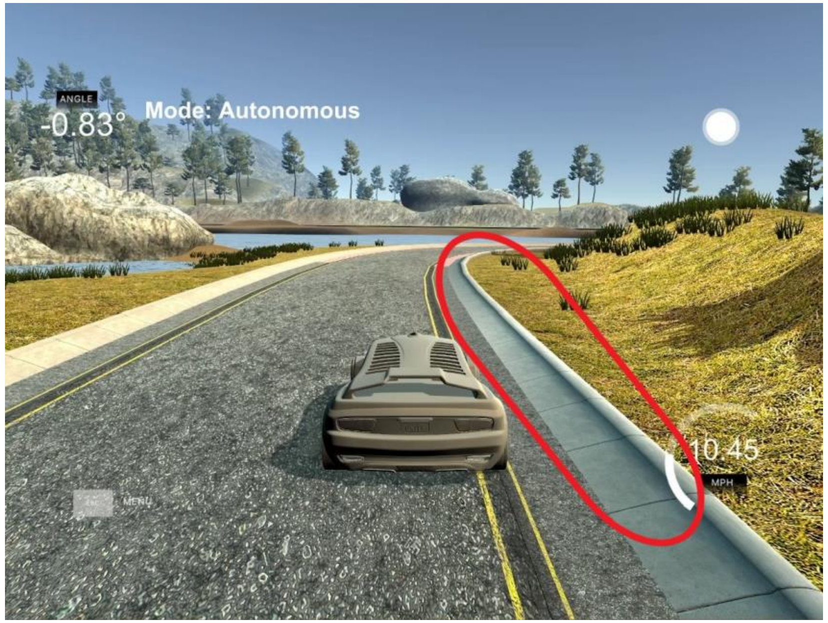
- The model converges fast in the first few epochs, which is efficient for self-driving in the simulator.
- After 20 epochs the models learns some wierd driving behavior
- Add a dropout layer to overcome overfitting
- Stop early to aviod learning unexpected behavior

DATA SHUFFLING

Illumimated Curbs



Curbs in Shadows



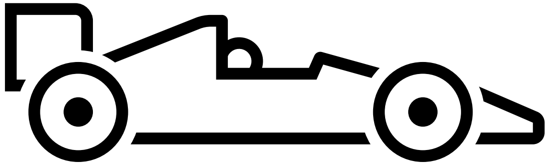
DATA SHUFFLING

- Without shuffling, the model learns the patterns only from the beginning of the route, yielding a bad performance in the other part of the road
- Shuffling can make the model learn from the entire route, increasing robustness.

RESULTS – DEMO VIDEO CLIP

- Reached a human drive rate of only 0.709%
- Most of the time, the car can position itself on the right course.
- It can detect the curvature of the road and drive along.
- The performance is not affected by lighting.

Full Demo Videos: Google Drive



CONCLUSIONS

CONCLUSION

- Implemented a self-driving car Udacity self-driving simulator
- Developed and tuned a CNN model
- Implemented data transformation
- SDC can deal with different road condition, degree of shadow and road marks
- Proved that the possibility for CNN models to perform the tasks of self-driving

THANK YOU