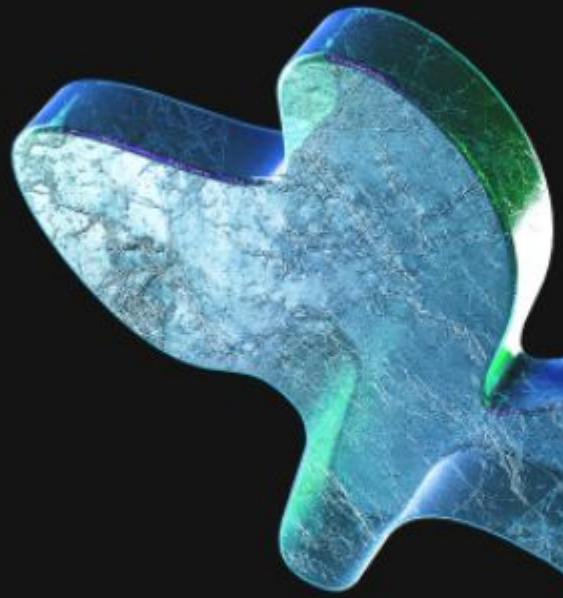
ENPM690 Final Project

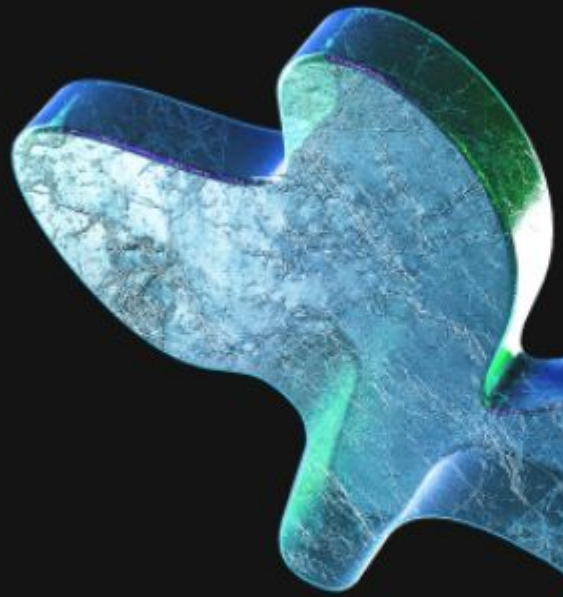
Deep Learning for Self-Driving Cars



Pratik Acharya 117513615
Abhijit Mahalle 117472288

Introduction

- Training a car to learn how to steer in a simulated environment using CNN.
- NVIDIA’s paper on End to End Learning for Self-Driving Cars
- Maps raw pixels from a single front-facing camera to steering command
- Learns internal representations to detect road features

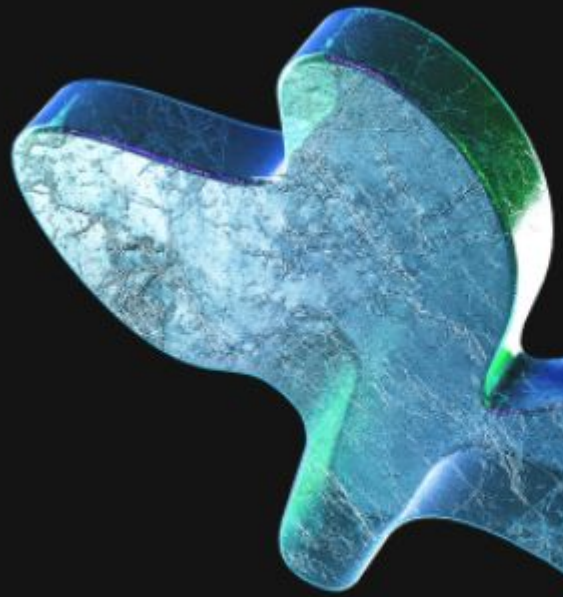


Simulator

- Udacity's self-driving car simulator
- Two modes: Training Mode, Autonomous Mode
- Two tracks

Programming

- Language: Python
- Framework: PyTorch



Data Collection

- Car was driven over two laps of Track 1
- Center, left, and right views were captured for each frame
- .csv file was generated; location of center, left, right views, steering angle, acceleration, brake, and speed for each frame

	A	B	C	D	E	F	G
26	C:\Users\Pratik A	C:\Users\Pratik A	C:\Users\Pratik A	0	0.048462	0	5.982447
27	C:\Users\Pratik A	C:\Users\Pratik A	C:\Users\Pratik A	-0.06222	0	0	5.95896
28	C:\Users\Pratik A	C:\Users\Pratik A	C:\Users\Pratik A	-0.38087	0	0	5.897681
29	C:\Users\Pratik A	C:\Users\Pratik A	C:\Users\Pratik A	-0.05994	0	0	5.877697

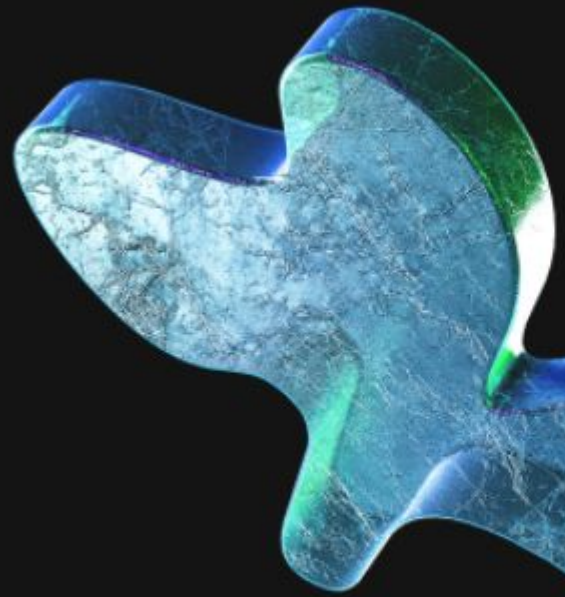
Data Pre-Processing

- Normalization:
  (x / 255) - 0.5
  Values between - 0.5 and 0.5

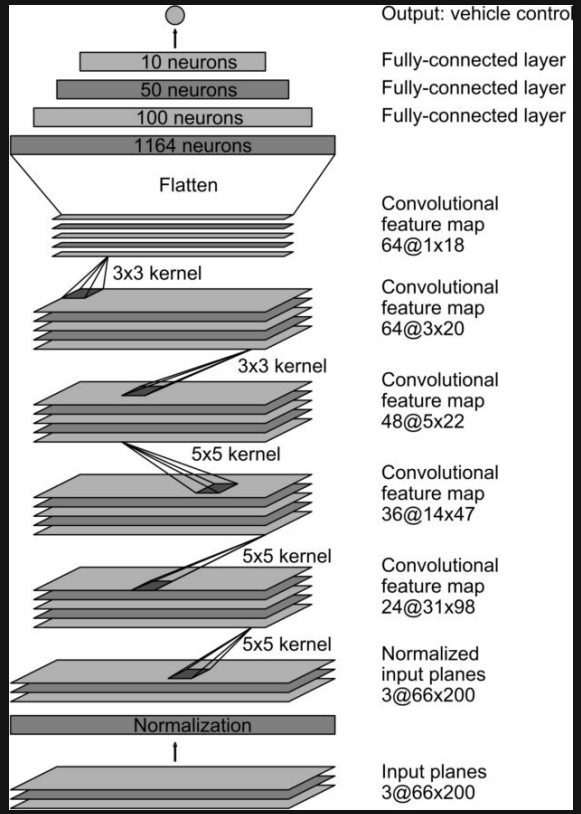
- Cropping

Data Augmentation

- Left/Right camera images; steering angle +- 0.4
- Random Horizontal Image Flipping



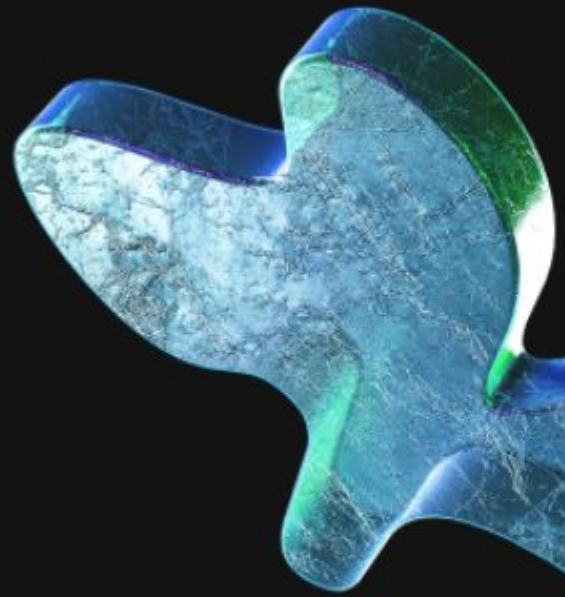
MODEL ARCHITECTURE



CONTROLLER

Simple PI Controller

Speed: 9 MPH

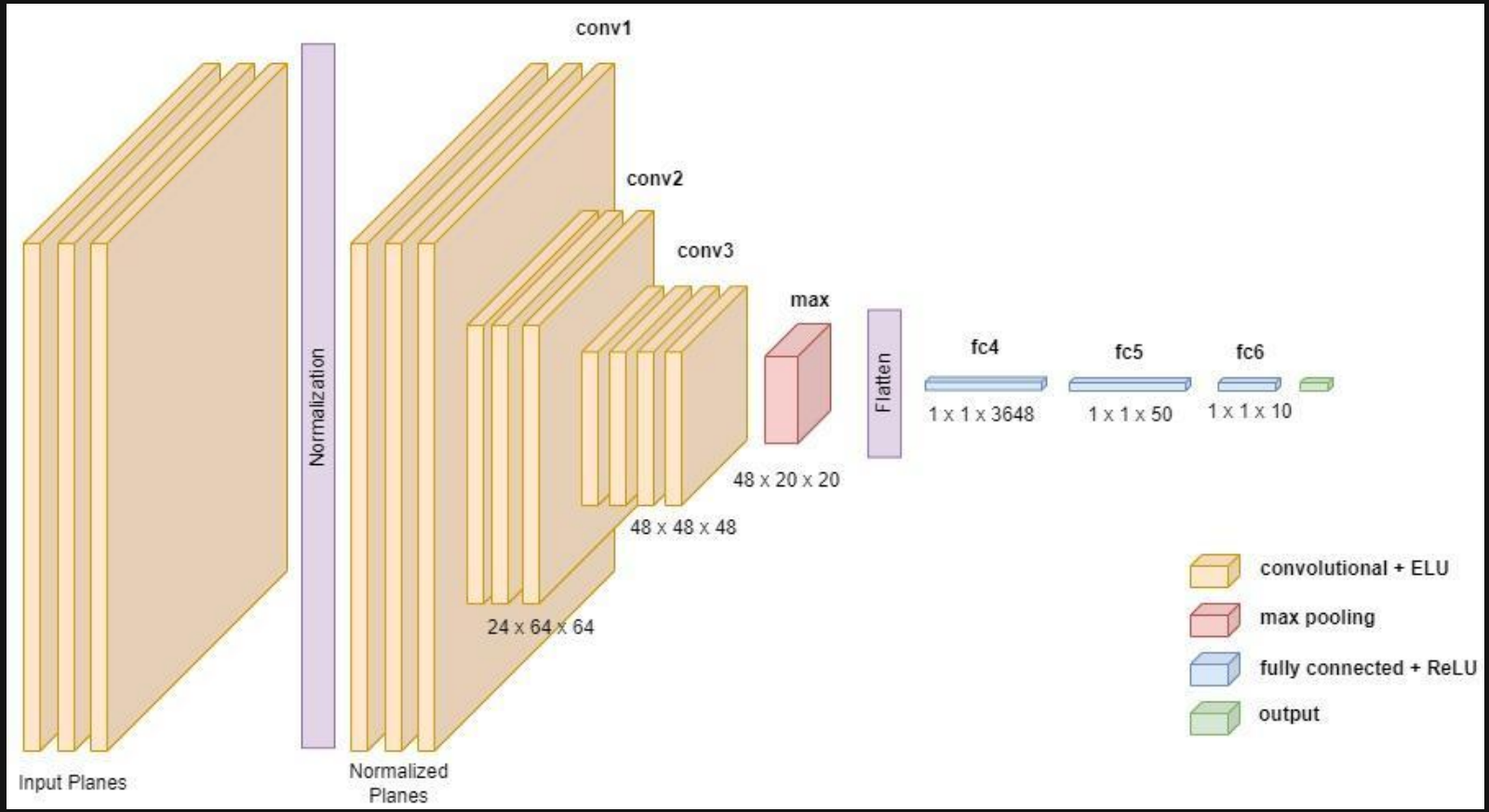


CHALLENGES

- Car drifting even on straight track

- Possible reasons:
  - Lack of sufficient data
  - Absence of Maxpool Layers

Simplified version of NVIDIA architecture

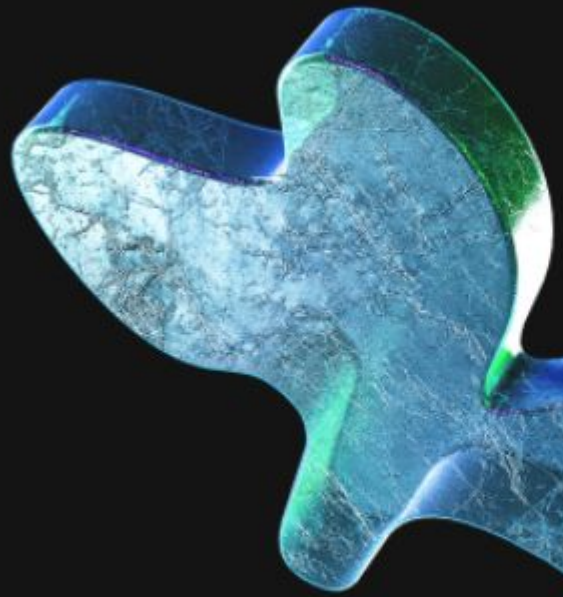


Results

Training Loss:

Left image: 0.041
Center image : 0.047
Right image: 0.048

Validation Loss: 0.026



HYPER-PARAMETERS

- Sample size: 24111
  Training: 19288 Validation: 4823
- Batch size: 32
- Optimizer: Adam
- Loss: Mean Square Entropy
- Epochs: 22
- Activation function: ELU
- Parameters: 120,000
- Connections: 15 million
- Training time: 22 minutes

DEMO


Steering Angle: -0.08823888 , Throttle: 0.07882739999999878
torch.Size([1, 88, 4, 10])
Steering Angle: -0.13276131 , Throttle: 0.07883629999999878
torch.Size([1, 88, 4, 10])
Steering Angle: -0.09778882 , Throttle: 0.07882909999999883
torch.Size([1, 88, 4, 10])
Steering Angle: -0.09779882 , Throttle: 0.07882139999999883
torch.Size([1, 88, 4, 10])
Steering Angle: -0.10941118 , Throttle: 0.07887279999999876
torch.Size([1, 88, 4, 10])
Steering Angle: -0.13107498 , Throttle: 0.07888199999999871
torch.Size([1, 88, 4, 10])
Steering Angle: -0.12107498 , Throttle: 0.07889119999999871
torch.Size([1, 88, 4, 10])
Steering Angle: -0.12107498 , Throttle: 0.07911039999999873
torch.Size([1, 88, 4, 10])
Steering Angle: -0.09884979 , Throttle: 0.07913819999999873
torch.Size([1, 88, 4, 10])
Steering Angle: -0.09887885 , Throttle: 0.07912959999999873
torch.Size([1, 88, 4, 10])
Steering Angle: -0.09887885 , Throttle: 0.07913899999999874
9.00

CONCLUSION

- CNN learn lane and road following without manual decomposition, semantic abstraction, path planning, and control.
- Learn meaningful road features from sparse training data
- Learns features without explicit labels during training.

REFERENCES

https://arxiv.org/pdf/1604.07316v1.pdf

https://developer.nvidia.com/blog/deep-learning-self-driving-cars/

http://yann.lecun.org/exdb/publis/pdf/lecun-89e.pdf

http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks. pdf