End-to-End Learning for Autonomous Vehicles using SVL

Team Bumblebee
Nitish Gupta
Xiangyu Chen
Kameron Ted Bielawski
Yiju Yang
Saharsh Gupta

Agenda

1. Background
2. Re-Implementation
3. Model Exploration
4. Demonstration
5. Conclusion & Future work

NVIDIA’s Attempt

Our Attempt

1. Background

1. Real World vs. Simulation

1. Advantages:

   Low-cost and flexible platform

   Realistic testing environment

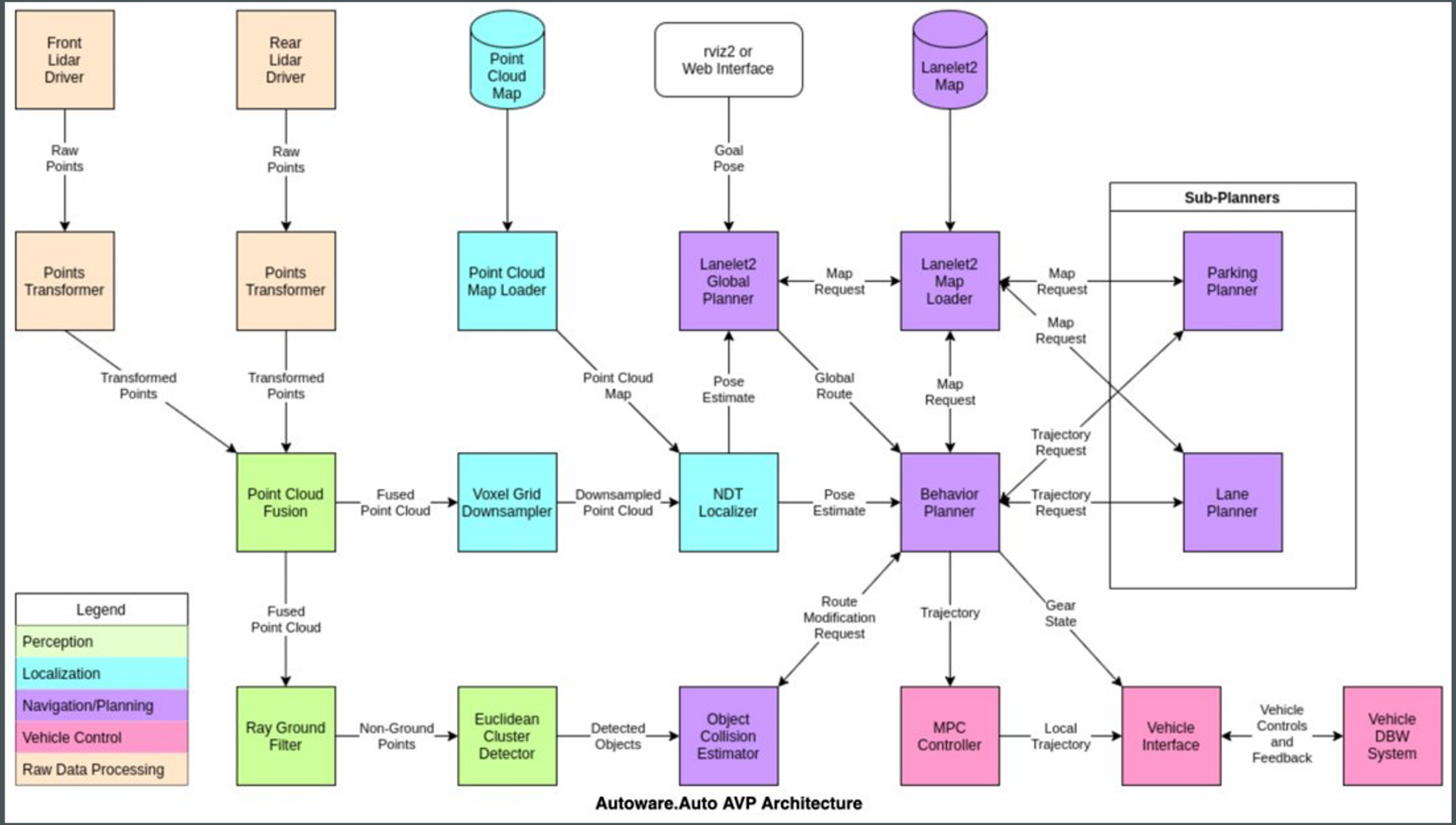
1. Constraints leading to unexpected driving behavior:

   Different weather conditions

   Road damage

   Low-visibility

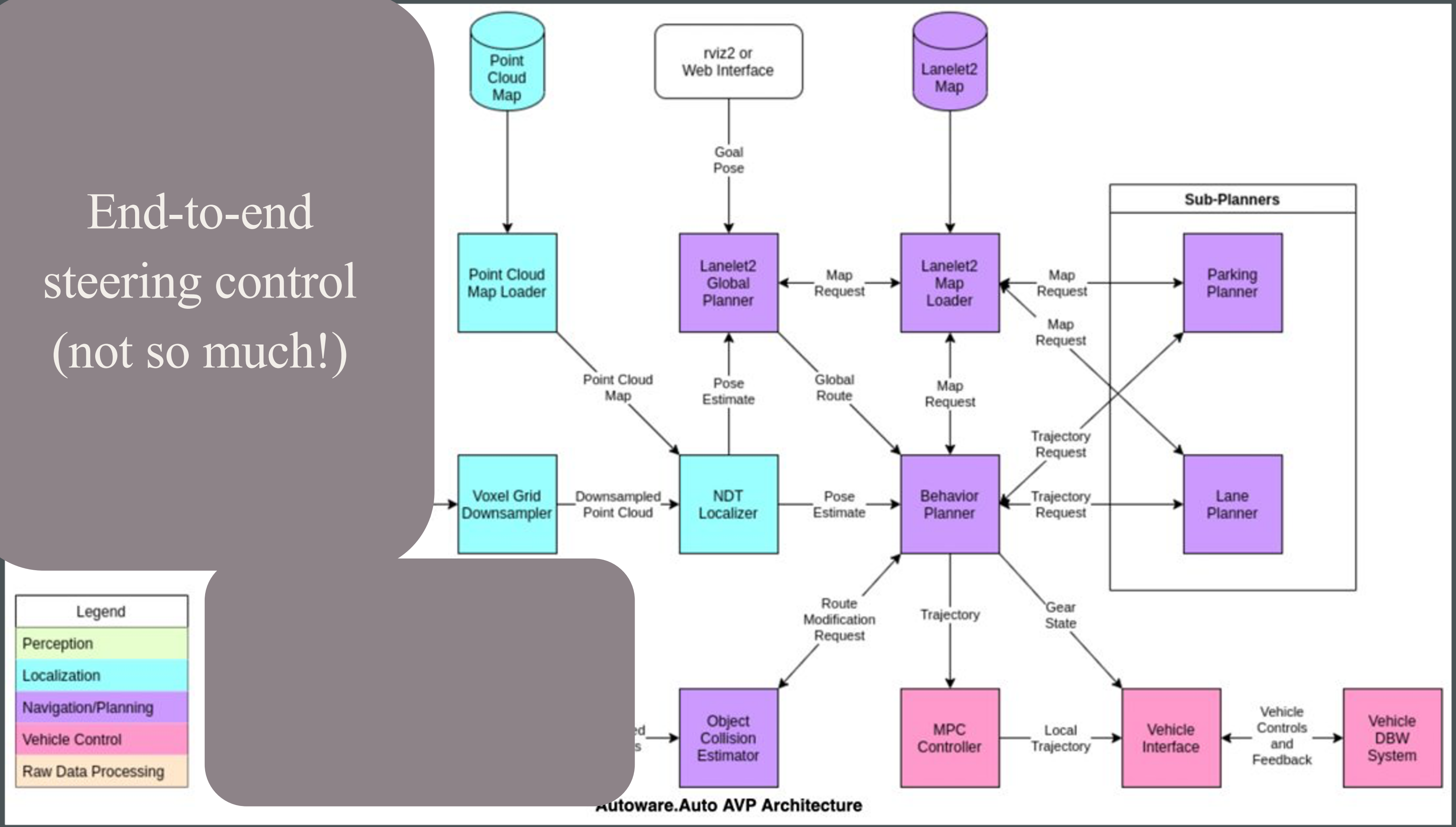
Standard Robotics Approach



Source: https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/avpdemo.html

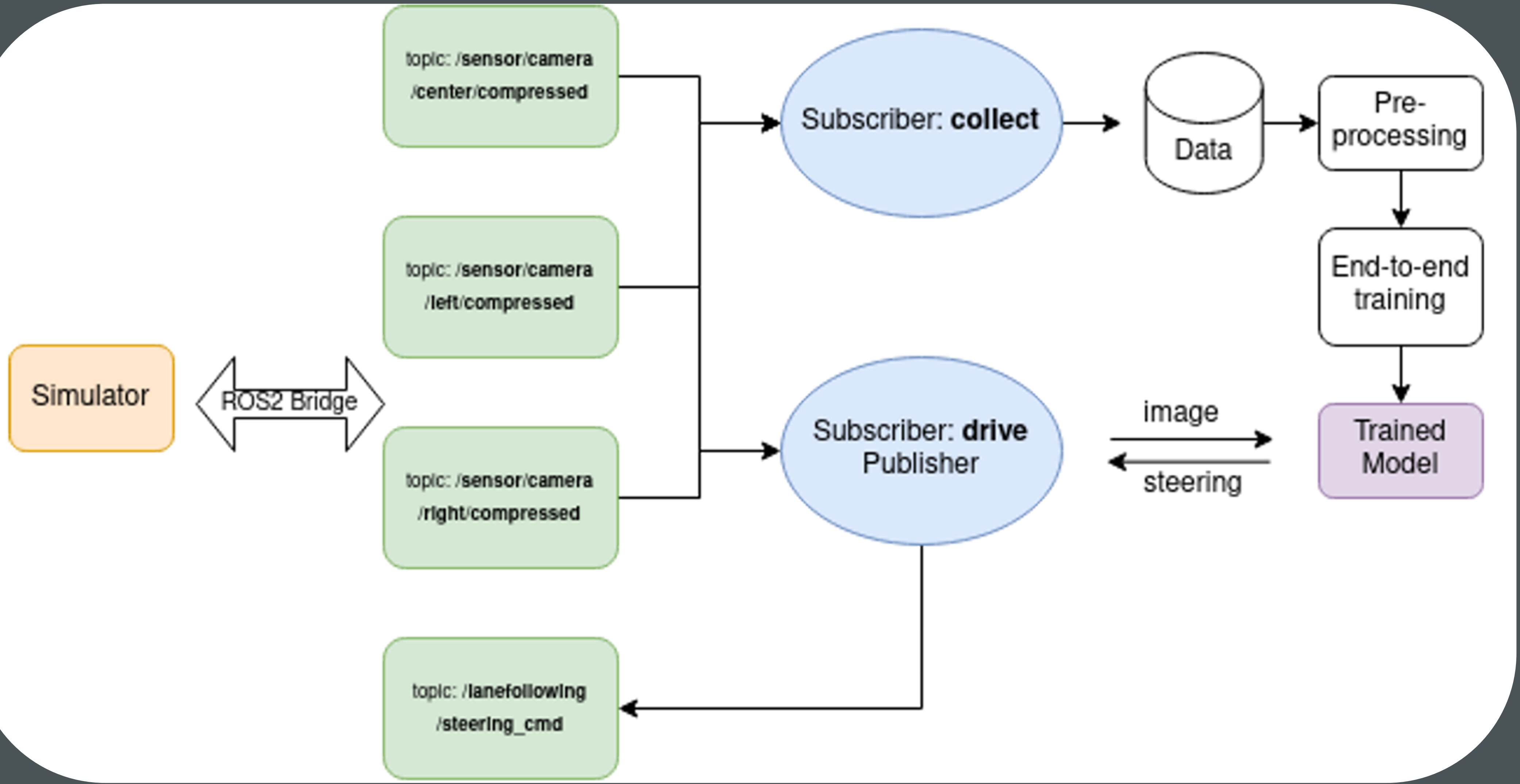
E2E Approach

End-to-end steering control (not so much!)



Source: https://autowarefoundation.gitlab.io/autoware.auto/AutowareAuto/avpdemo.html

2. E2E Architecture



Data Acquisition

1) Map:

   4 different maps (San Francisco, BorgesAve, CircularPath, Lane-less road)

2) Data:

   a. Camera feeds: left, right, center

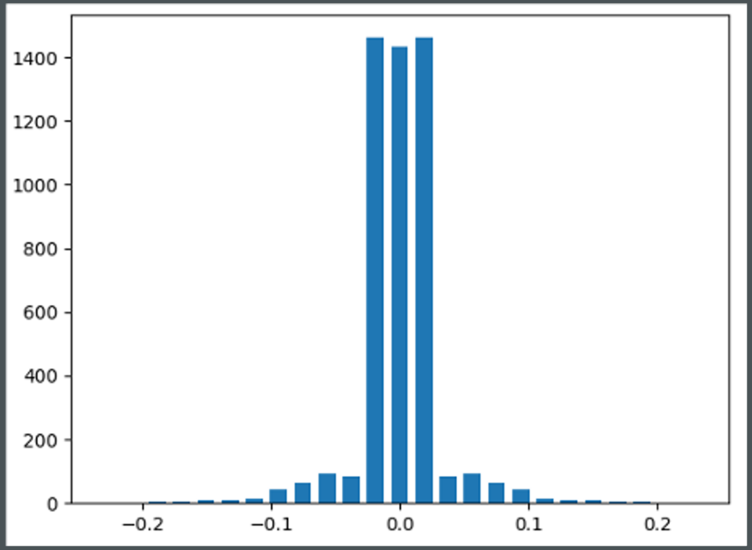
   b. Corresponding steering angle values

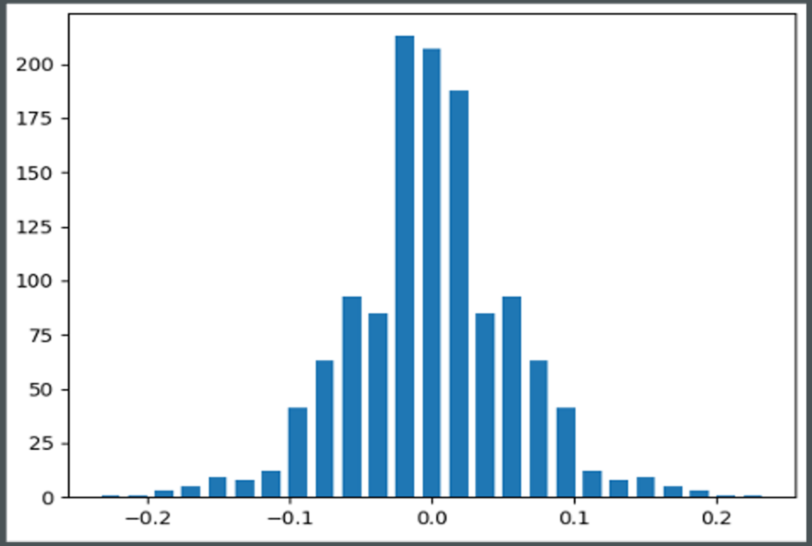
3) Drive speed: constant speed

4) Time: 10+ hours of driving

Preprocessing 1: Data Augmentation

- Data addition:
  - Flipping images
  - Negating corresponding steering angles
- Histogram balancing
  - Randomly resampling to the mean to remove bias





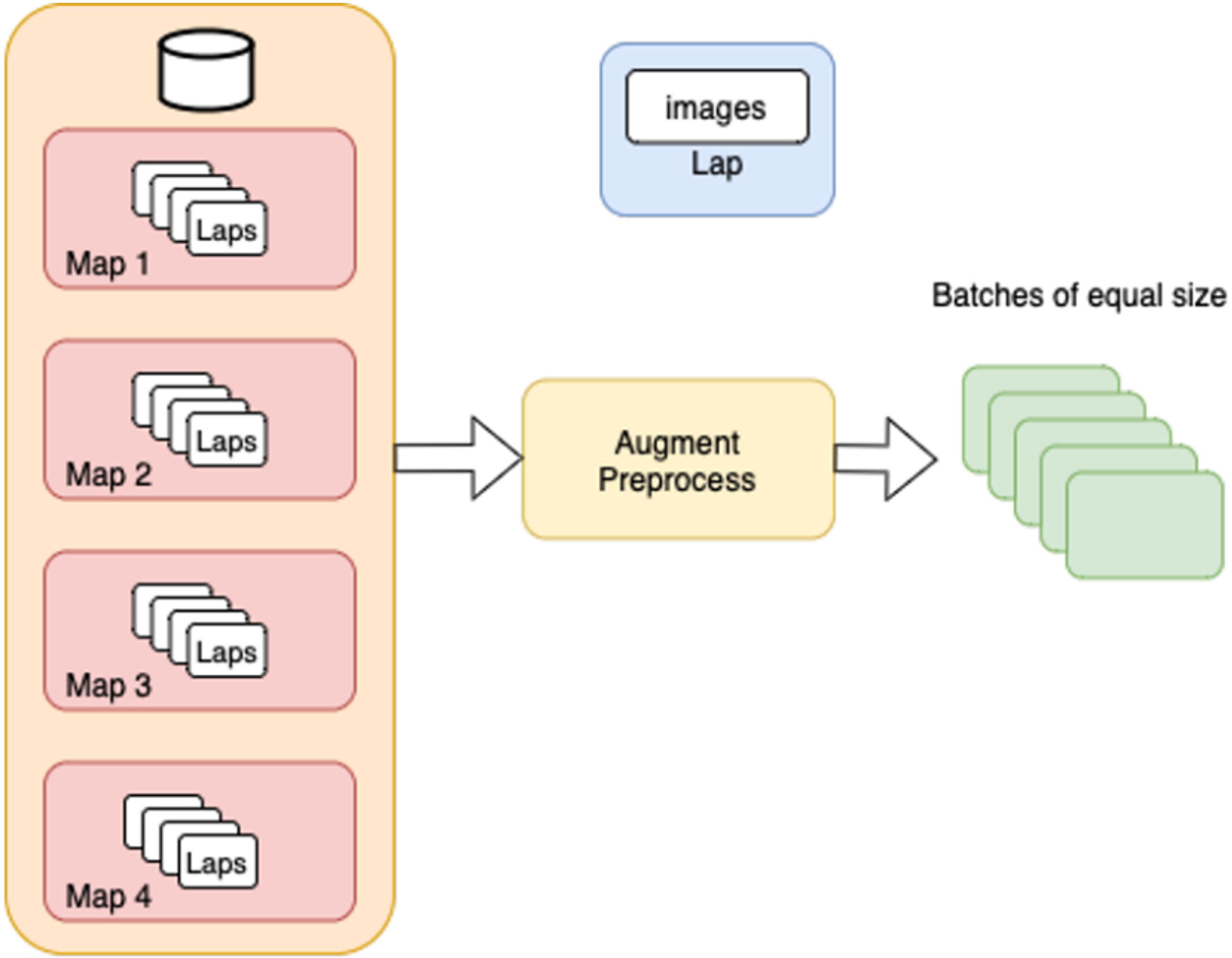
Preprocessing II: Crop and Scale

- Cropping
  - Removal of irrelevant information (sky, hood, etc.)
- Scaling
  - Reducing resolution
  - Less data-intensive

1920x1080x3

780x222x3

Preprocessing III : Batching



Training

Baseline: NVIDIA Model

- Covnet : 9 (5 conv+4 dense)
- # of Parameters : 250k
- Inference time : 10ms
- FPS : 100 FPS
- Successfully completed laps on all maps

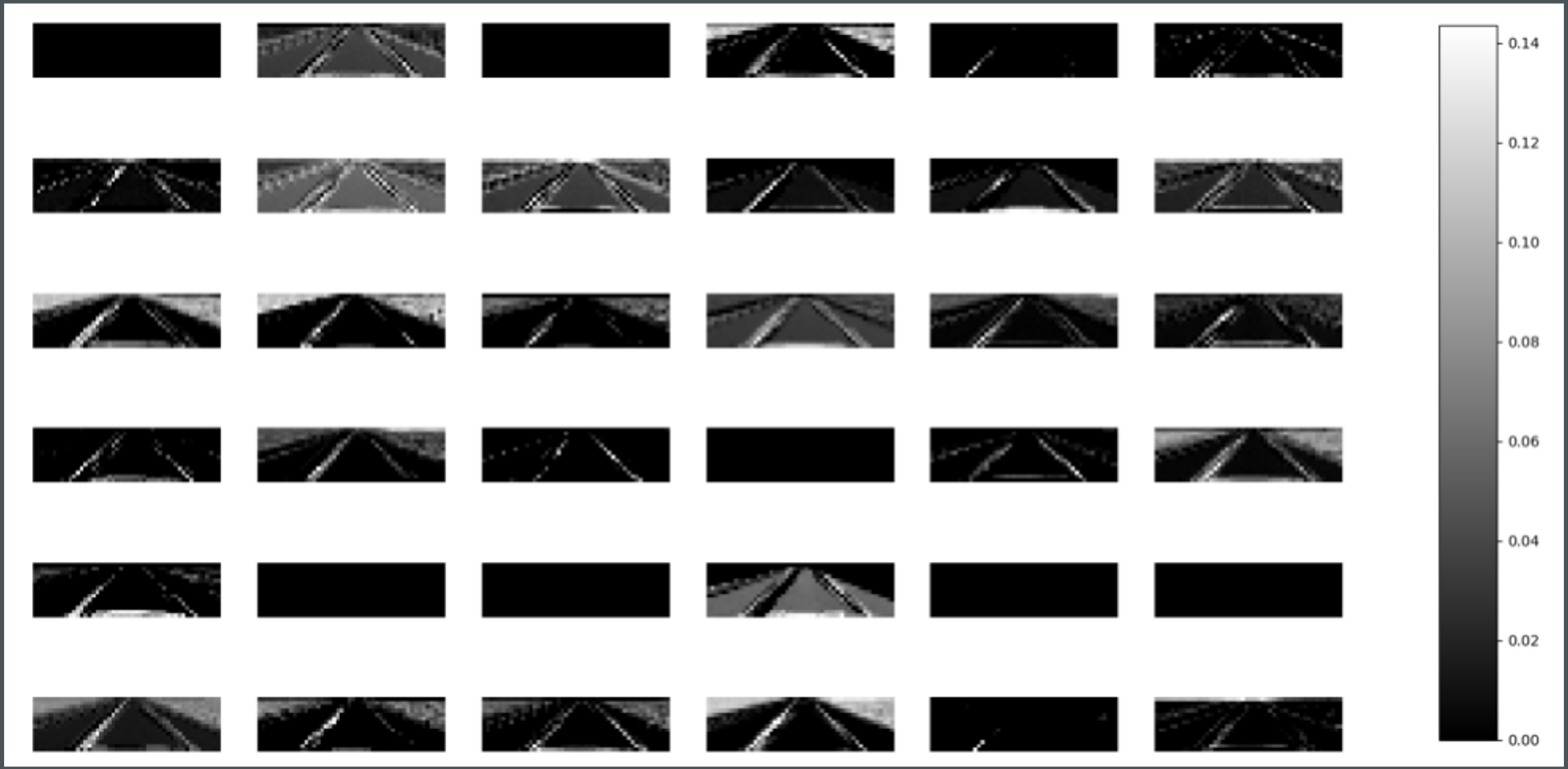
Convnet training details:

- Optimizer : Adam
- learning rate : 10-4
- Epochs : 10
- Batch size : 32 or 128
- Callbacks : Early stopping
- GPU : GTX 1080 8GB

What matters?

- Lower number of parameters leads to lower inference time
- Validation MSE < 0.002

Training: Feature Maps



4. Network Exploration

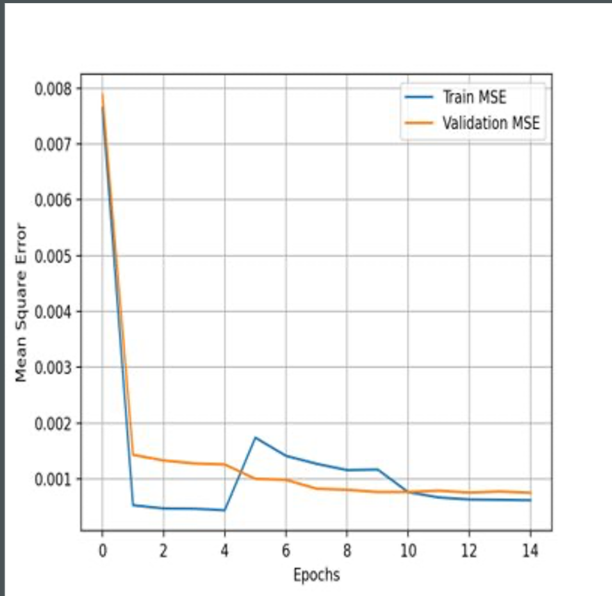
- Following the principles we obtained in simple Convnet models, e.g. # of params, we choose networks as below:

backbones	# of params	FPS	comment
Convnet v1 (NVIDIA)	0.25 m	100	🐝
Convnet v2	70 m	20	☠️
Simple Resnet	0.95 m	24	☠️
RNN	31 m	15	🐝
Shufflenet v1	3.8 m	-	☠️
Shufflenet v2	4.0 m	-	☠️
Resnet10	4.9 m	50	☠️
Resnet18	11.2 m	35	☠️

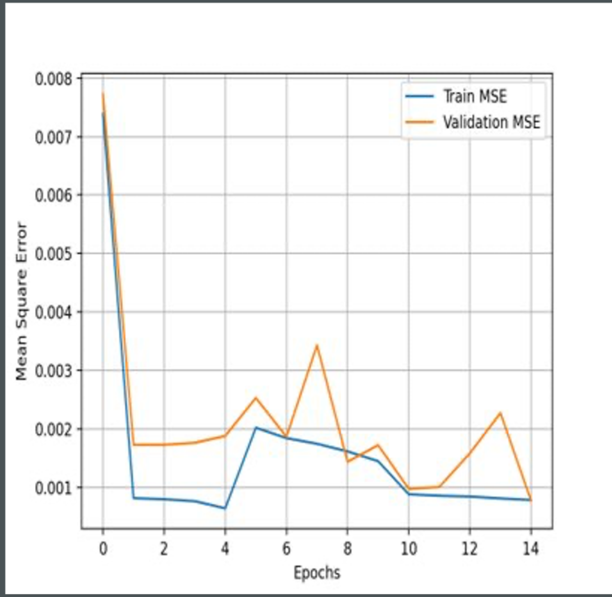
🐝 – completed Lap
☠️ – Accident

4. Network Exploration

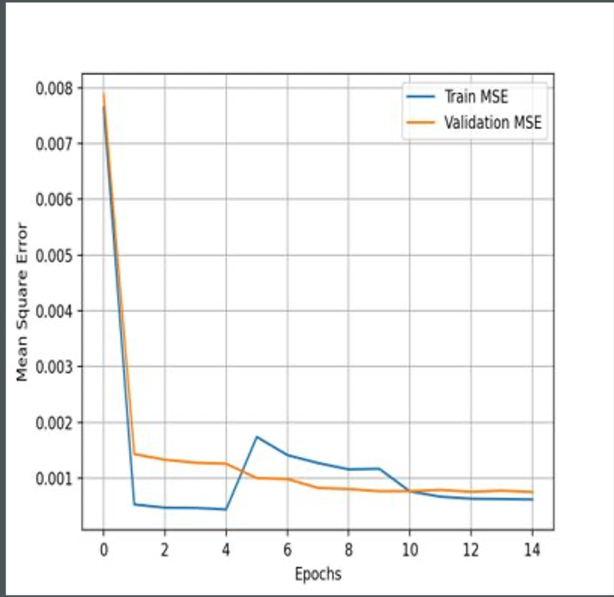
- Training details:



Convnet v1 (NVIDIA)



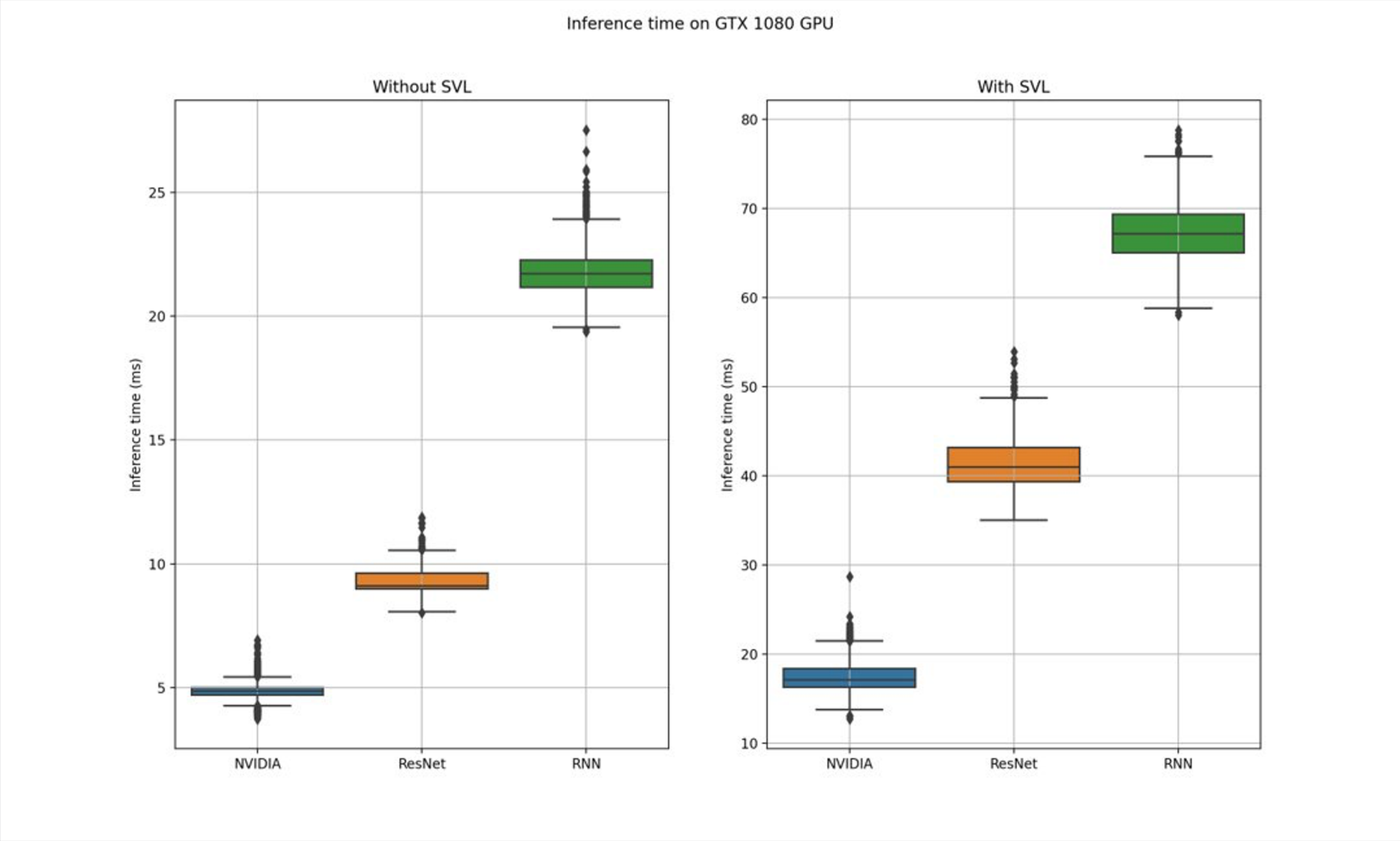
Simple Resnet



RNN

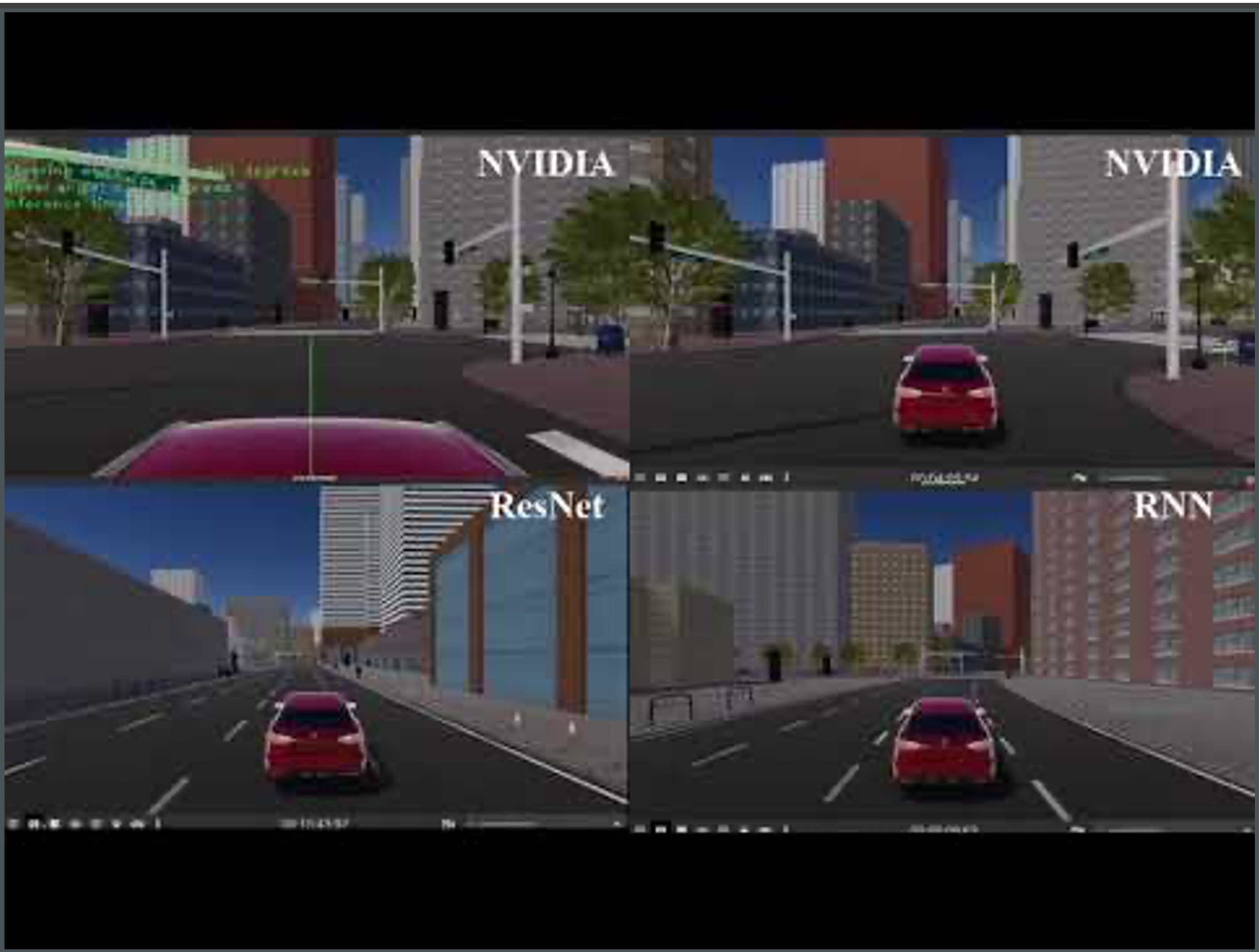
4. Network Exploration

- Inference time:



Demos

Demos



5. Conclusion & Future work

- Multiple maps – one model
- RNN sliding window
- Resnet50 with cluster
- Training with traffic
- Predicting throttle

REFERENCES

1. Bojarski, M., Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End Learning for Self-Driving Cars. *ArXiv, abs/1604.07316*.
2. www.svlsimulator.com
3. www.ros.org
4. J. Zhou, X. Hong, F. Su and G. Zhao, "Recurrent Convolutional Neural Network Regression for Continuous Pain Intensity Estimation in Video," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016, pp. 1535-1543, doi: 10.1109/CVPRW.2016.191.
5. Eraqi, H.M., Moustafa, M.N., & Honer, J. (2017). End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies. *ArXiv, abs/1710.03804*.

THANK YOU