Motion Prediction for Autonomous Vehicles

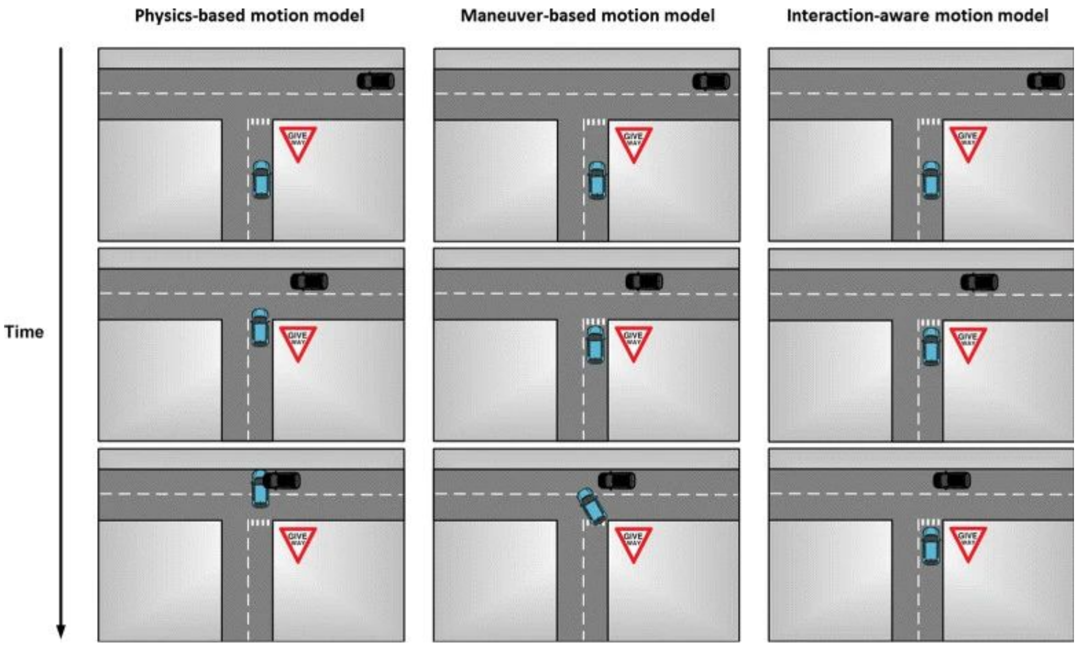
Aaron Guan, Yi Gu, Wanzhi Zhang

Motivation

- Related to self-driving cars
- To detect dangerous situation and react accordingly to ensure safety
  - To predict the likely evolution of the current traffic situation
  - To assess how dangerous that future situation might be, which is a support for decision making
- Challenges:
  - Vehicles' behavior on the road will affect the behaviour of other vehicles.
  - Sudden changes and environment conditions will affect the behaviour of vehicles.

Prior Work - Prediction Model

- Physics-based model
  - Only depend on the laws of physics
- Maneuver-based model
  - Consider the maneuver that the driver intends to perform
- Interaction-aware model
  - Take the interdependencies between vehicles’ maneuvers



Prior Work - Related method

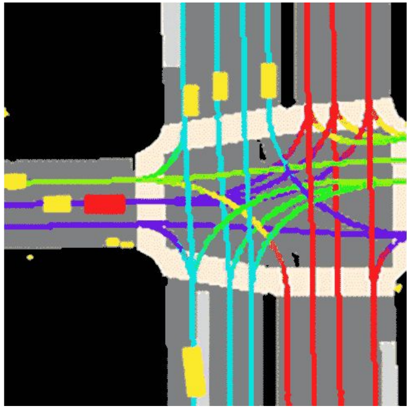
- Convolutional Neural Networks
  - Six layer CNN with convolution and fully connected layers are used to predict the intention of surrounding vehicles.
  - A convolution-deconvolution architecture, introduced in [2], is used to predict vehicle behaviour.
  - First, two backbone CNNs are used to extract the features of lidar data and rasterized map [3] separately. Then three different networks are applied to the concatenation of extracted features.

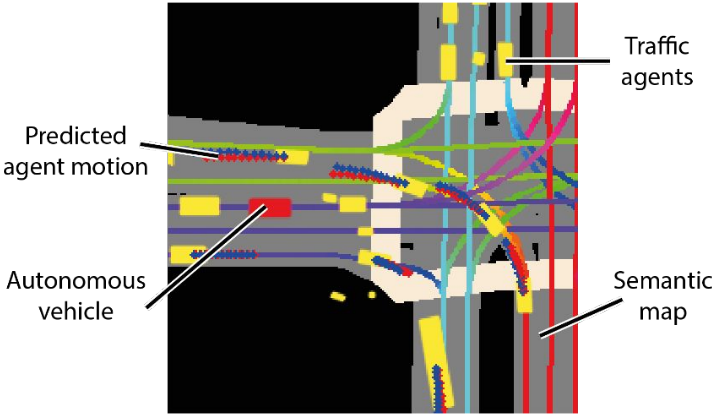
Dataset

- Total data set size: 1,118 hours / 26,344 km / 162k scenes
  - Training set size: 928 hours / 21,849 km / 134k scenes
  - Validation set size: 78 hours / 1,840 km / 11k scenes
  - Test set size: 112 hours / 2,656 km / 16k scenes

- Scene length: 25 seconds
- Total # of traffic participant observations: 3,187,838,149
- Average # of detections per frame: 79
- Labels: Car: 92.47% / Pedestrian: 5.91% / Cyclist: 1.62%

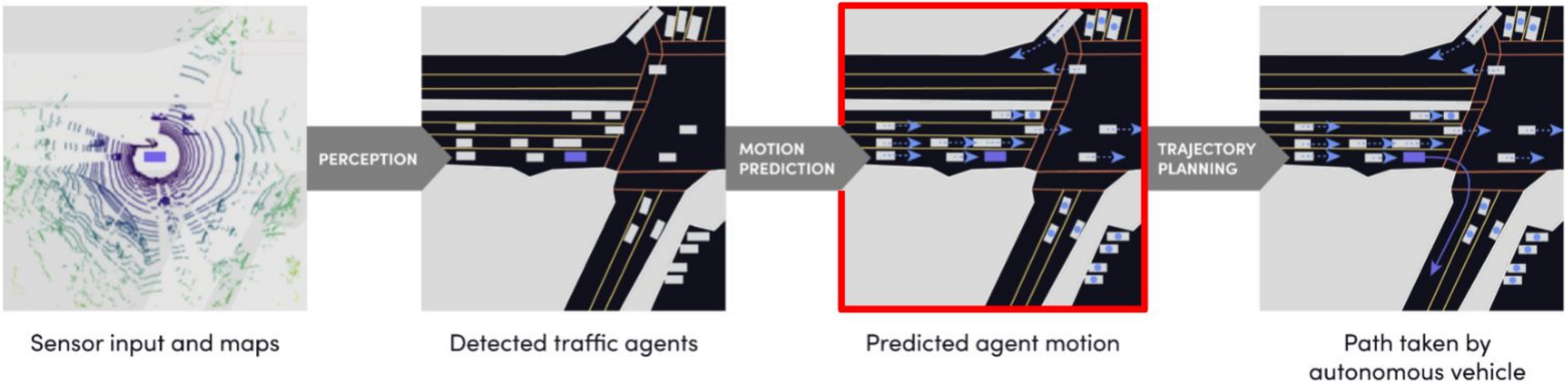
- Semantic map: 15,242 annotations / 8,505 lane segments
- Aerial map: 74 km^2 at 6 cm per pixel



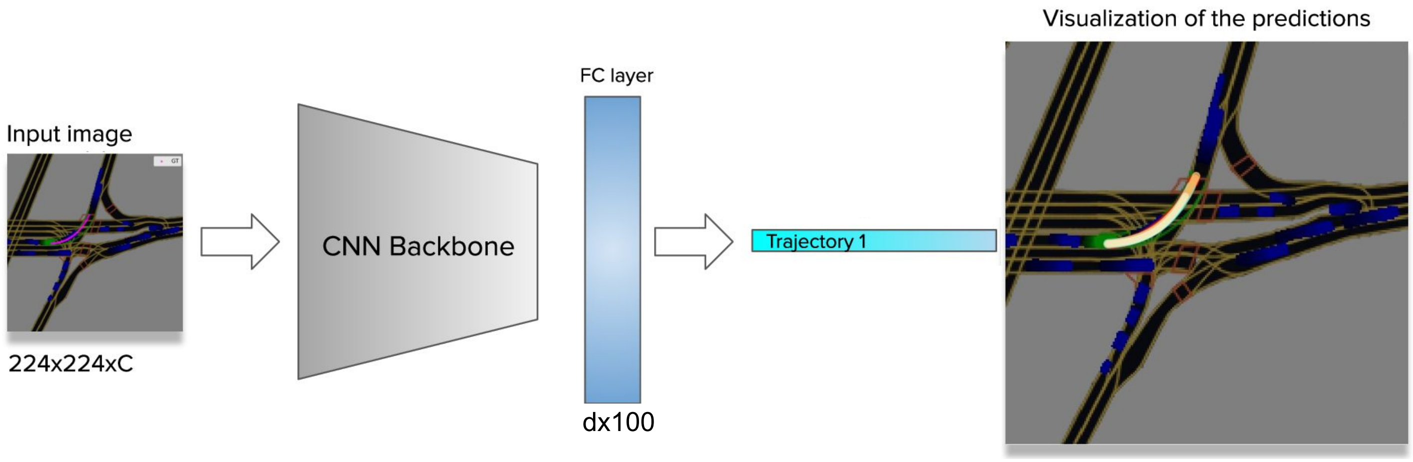


Idea

- Predict the trajectory of the agent for the next 5s in future given their historical 5s positions
- Trained CNN regression pipeline with images



Architecture



- **Input Channels:** 3 + (50 + 1) * 2 = **105**
  - **3** for RGB image.
  - **50** history time steps and **1** current (each time step has 0.1s interval).
  - Every time step is represented by **2** channels: (1) mask representing the location of the current agent (2) mask representing all other agents nearby.
- **Output Channels:** 50 * 2 = **100**
  - 50 future time steps and their X, Y positions
- **CNN Backbone:** ResNet 28 and Xception 41

Loss

Loss Function: Negative Log-Likelihood

$$\mathbf{GT} = [\overbrace{(x_1, y_1), \ldots, (x_{50}, y_{50})}^{\text{Ground truth coordinates}}]$$

$$\mathbf{hypothesis}_k = [(x_1^{(k)}, y_1^{(k)}), \ldots, (x_{50}^{(k)}, y_{50}^{(k)})],\ k = 1, \ldots, K$$

$$L = -logP(\mathbf{GT}) =$$

$$= -log \sum_k \overbrace{c^k}^{\text{Confidences}} \prod_t \mathcal{N}(\mathbf{GT} | \overbrace{\mu = \mathbf{hypothesis}_k}^{\text{Predicted hypothesis \#k}}, \Sigma = E)$$

$$= -log \sum_k c^k \prod_t \mathcal{N}(x_t | \mu = \bar{x}_t^{(k)}, \sigma = 1) \mathcal{N}(y_t | \mu = \bar{y}_t^{(k)}, \sigma = 1)$$

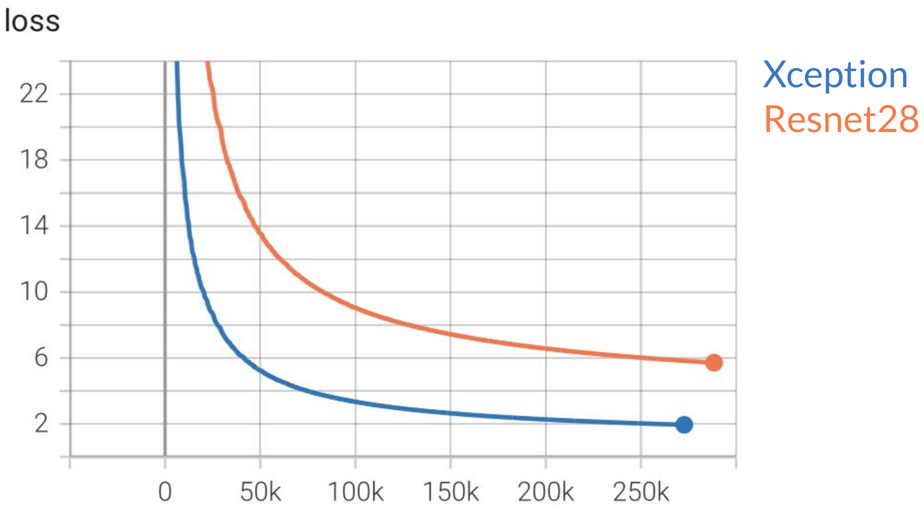
Predicted coordinates

$$L = -\log \sum_k e^{\overbrace{\log(c^{(k)})}^{\text{Confidences}} - \frac{1}{2} \sum_t (\bar{x}_t^{(k)} - x_t)^2 + (\bar{y}_t^{(k)} - y_t)^2}$$

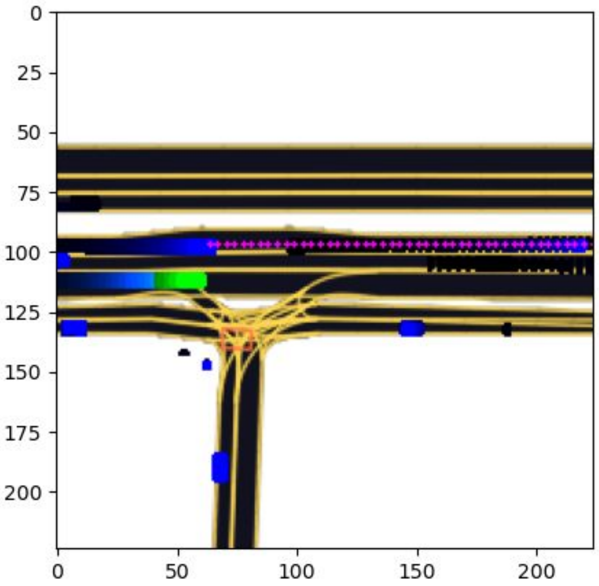
Ground truth coordinates

Results

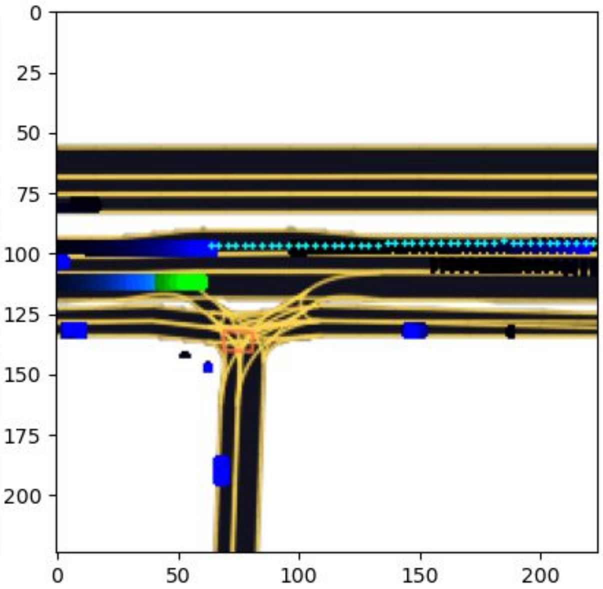
- Image Size: 224 x 224
- Batch Size: 32
- SGD optimizer with learning rate 0.001
- 1 Tesla T4 GPU trained with ~280000 iterations for about 100h on AWS



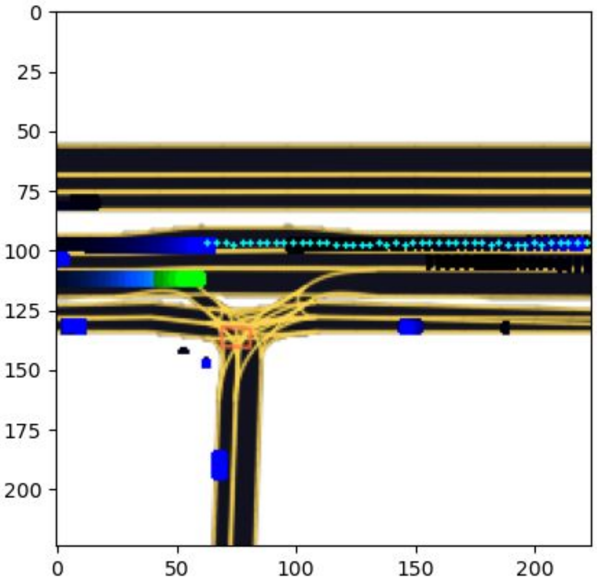
Results



Frame no. 209299
Ground Truth

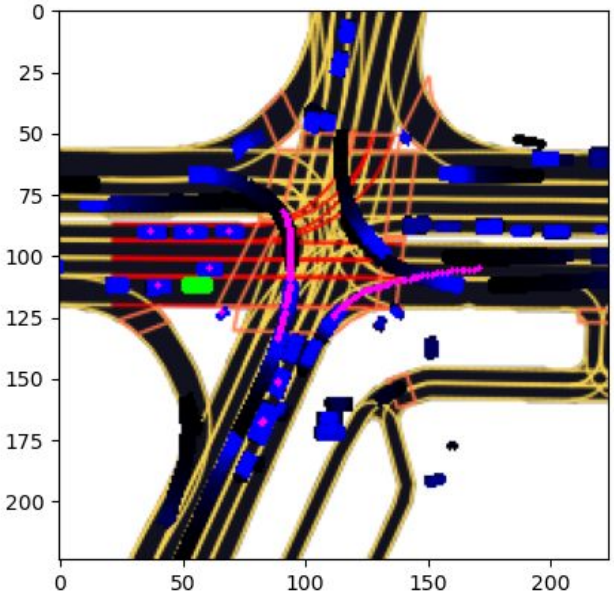


ResNet Prediction

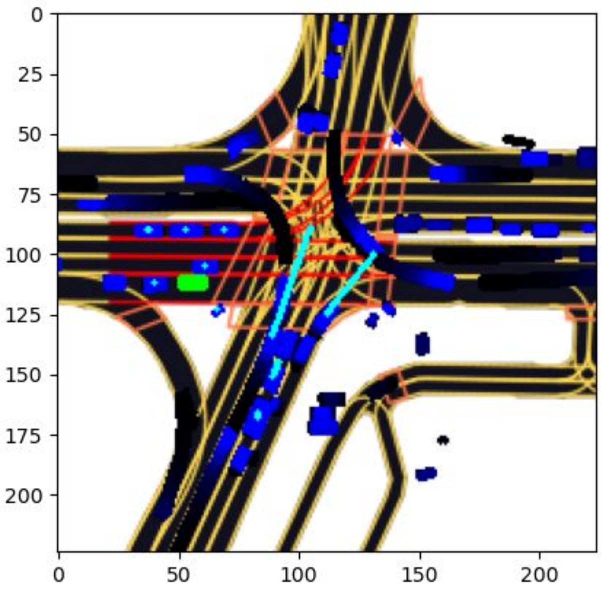


Xception Prediction

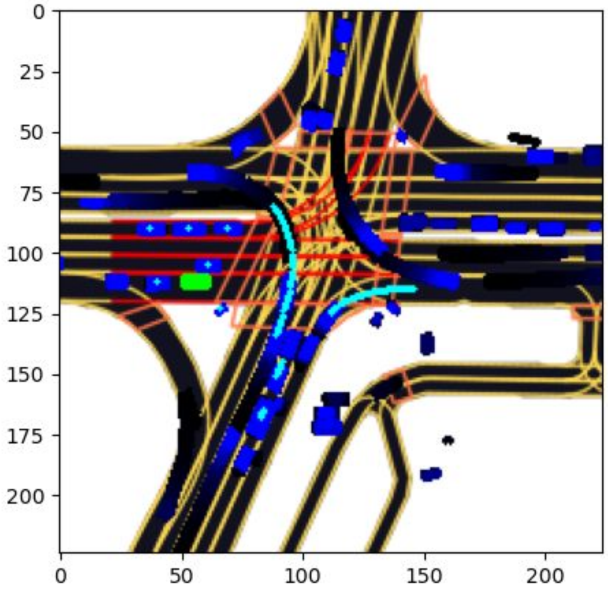
Results



Frame no. 1576699
Ground Truth

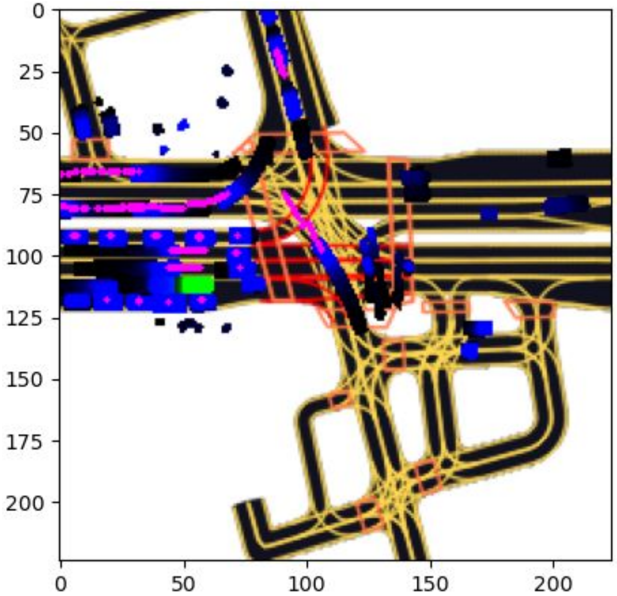


ResNet Prediction

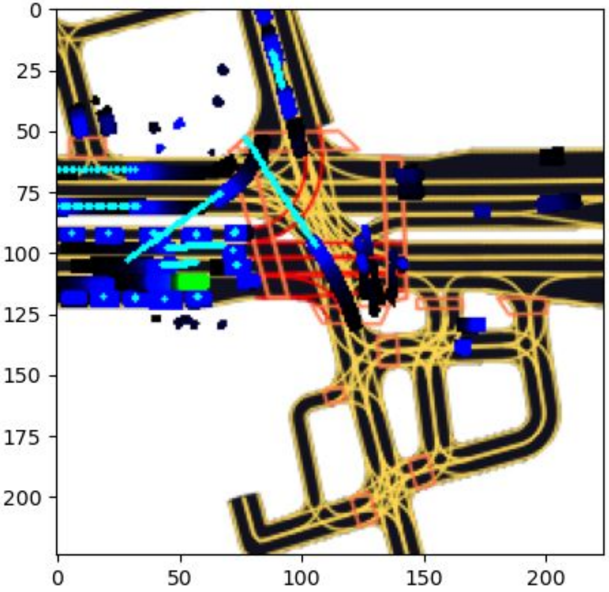


Xception Prediction

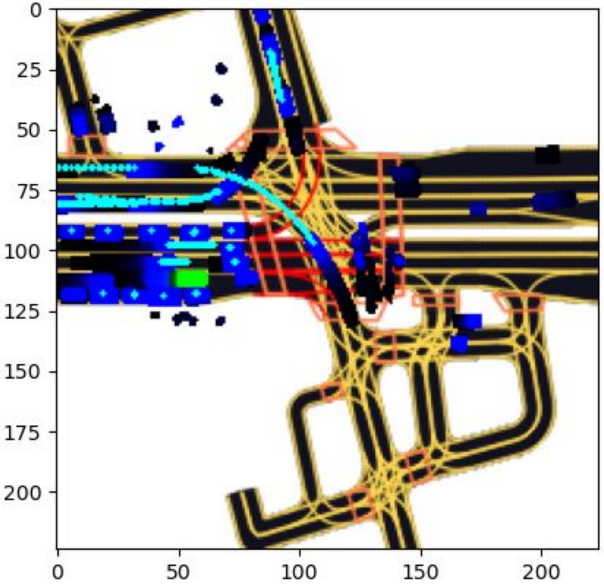
Results



Frame no. 695299
Ground Truth

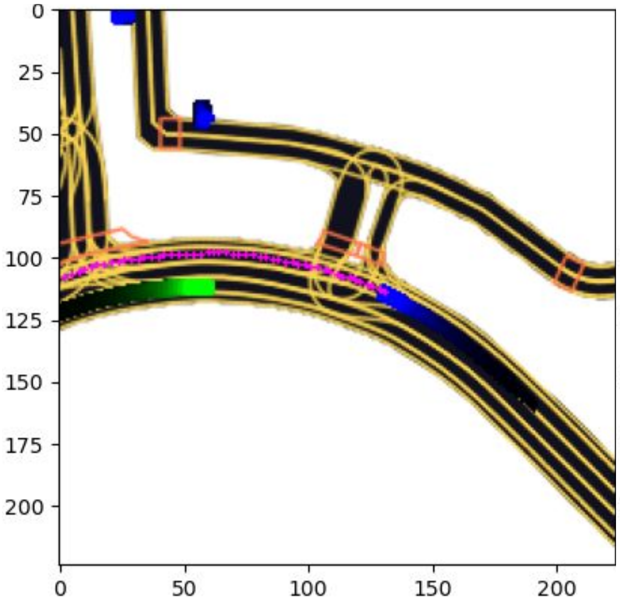


ResNet Prediction

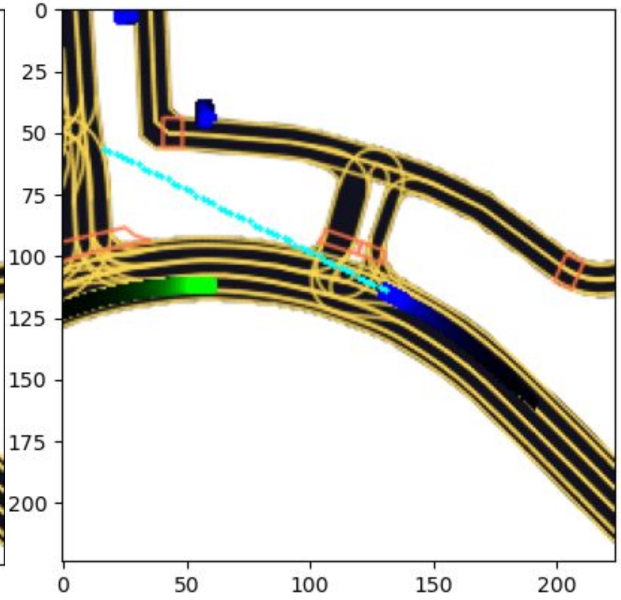


Xception Prediction

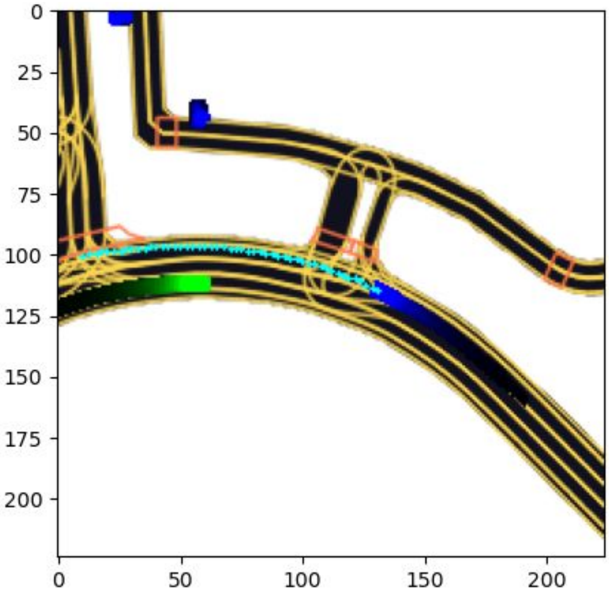
Results



Frame no. 695299
Ground Truth



ResNet Prediction



Xception Prediction

Future Work

- Ensemble more models
- Generate more possible trajectories with their confidence scores
- Try L1/L2 loss functions and compare with NLL.
- Predict the ego car’s future trajectory with agents’ predicted trajectories

Reference

[1] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp.353–359.

[2] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, 29–31 Oct 2018, pp. 947–956. [Online]. Available: http://proceedings.mlr.press/v87/casas18a.html