Vehicle Motion Prediction with CNN

Stanford University

Yaqi Zhang (yaqiz), Billy Wan (xwan), Wenshun Liu (wl88)

Introduction

- Vehicle motion detection is useful in analyzing pre-recorded vehicle videos, providing knowledge of at-the-time vehicle status and serving as a fundamental element for other downstream tasks such as driver intention inference.
- Most existing speed prediction are based on video footage recorded from still camera. They are mostly image-processing based techniques that utilized a particular subset of information such as the extend of motion blur (which falls short when speed is low and motion blur is minimal) and motion detection with pixel-wise difference.

Problem Statement

- In this project, three motion-related properties (forward speed, forward acceleration, and angular velocity) are predicted using inputs of dense flow matrices, object detection masks, and RGB values obtained from each frame.
- Each combination of inputs (Fig.1) is evaluated against a baseline 2-layer CNN and a AlexNet architecture model.
- Mean-squared error (MSE) loss is used to optimize the model and conduct quantitative assessments. Predicted speed and direction are also visualized on top of each testing frame for qualitative evaluation against human intuition.

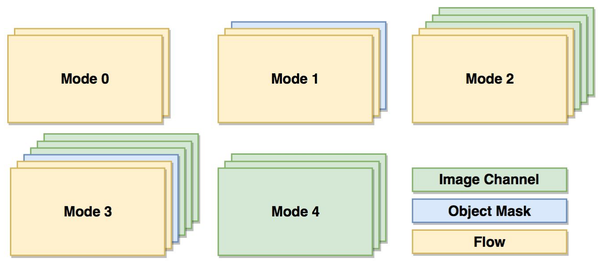

Fig. 1 Input structure

Dataset

- 23 videos from KITTI dataset of front-view camera recording at 10 Hz with a resolution of 1242×375.
- ⅓ of frames are sampled from all video to reduce storage pressure.
- Artifacts: dimension mismatch between different videos

Models

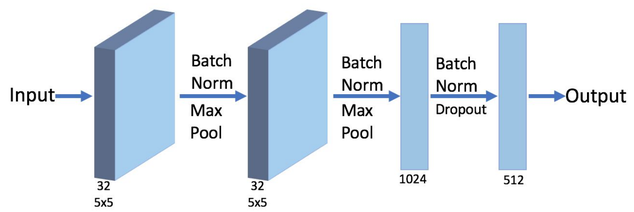

Fig. 2 Baseline CNN Architecture

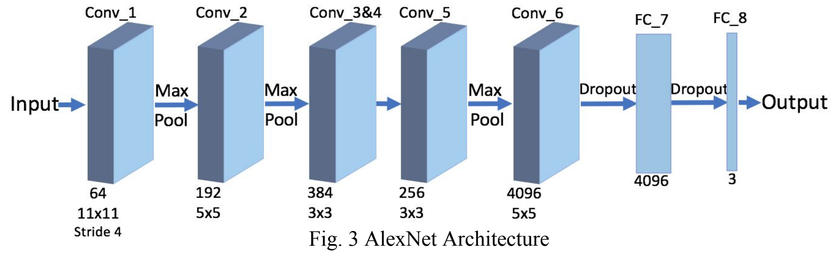

Fig. 3 AlexNet Architecture

Fig.2 and Fig. 3 show architectures of our baseline CNN model and AlexNet model. Since model input is not always an image, our models are trained from scratch.

- **Baseline CNN** : 2x conv-relu-batch_norm-pooling layers, 2x affine-relu-batch_norm-dropout layers with a dropout rate of 0.4, and a final affine layer to produce the same output. The Adam optimizer is used with a constant learning rate of 0.001.
- **AlexNet** : the architecture proposed by Alex Krizhevsky, and later refined to take advantage of parallelization of CNNs.

We use MSE loss, which measures on average how close the predictions are to the ground truth labels, to evaluate the model's predictions. Given predictions y and ground truths y for a batch size of N, the MSE loss L is computed using $L = \frac{1}{N}\sum_{i=1}^{n}(\hat{y} - y)^2$

Results and Discussions

- Best performance is achieved when all types of information are used. This meets our expectation in that information about relatively moving objects (object mask) and road (rgb) in addition to optical flow is able to aid speed prediction.
- Despite its more sophisticated model, AlexNet is unable to outperform the baseline model at the moment. More complex model is harder to train and sensitive to weights initiation. This suggests the need for more extensive hyperparameter search and better ensembling methods.
- Loss curves indicate no strong overfitting (controlled through param tuning such as dropout, weight/lr decay). Some modes could be trained for longer epochs for better convergence.
- Visualized results roughly match the loss we see from quantitative results.
- Angular prediction error is in general much worse than the others and the prediction is skewed towards believing vehicle is making right turn. Currently we train a single model to predict all interested properties and angular velocity contributes to a small amount to overall loss due to its small magnitude in radius.

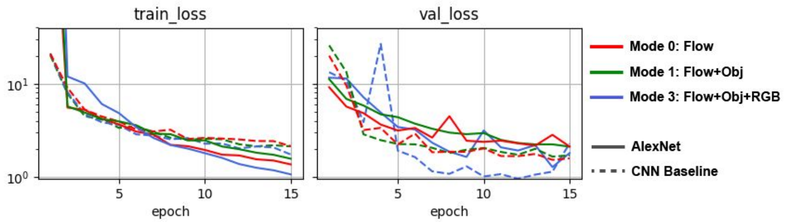

Fig. 4 MSE Loss decay over epochs for different modes

Best MSE Loss	Train	Val
Baseline	4.26	3.15
AlexNet	2.122	3.56

Table 1. Best MSE Loss

Conclusions

- We trained a vehicle motion prediction model with different CNN architectures using multiple types of information from videos, and were able to achieve ~3 total MSE for 3 properties of interest.

Future Work:

- Train separate models, one for each motion property.
- More meaningful inferences can be made using the motion predictions combined with other environmental information.
- With larger hardware storage support, more data and larger batch size could achieve better performance.