CNN, CNN Encoder-RNN Decoder, and Pretrained Vision Transformer for Surrounding Vehicle Lane Change Classification at Future Time Steps

Neha Konakalla[1], Avrum Noor[2], Josh Singh[3]

Department of Computer Science, Stanford University

[1]nkona@stanford.edu, [2]avrum@stanford.edu, [3]jsingh5@stanford.edu

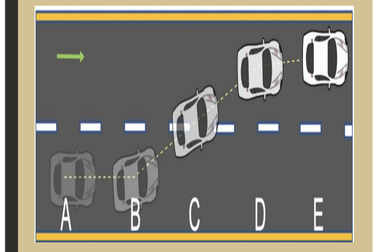
Stanford Computer Science

Introduction

The **enhanced safety benefits** that autonomous vehicles provide over human drivers have led to the popularity of autonomous vehicles research and application.

Lane change (LC) is a critical manoeuvre for the safety of vehicles in highways.

While the current state of autonomous vehicles have shown much performance gains at performing actions like lane changing, stopping, and accelerating, much work is still needed when it comes to **reliably predicting lane changing of surrounding vehicles.**



We propose a **CNN Baseline, CNN-encoder RNN-decoder**, and a **fine-tuned pre-trained ViT model** and demonstrate their performance on L**C classification at future time steps.**

Stanford University

Problem Statement

Input: Video from front of a car which is then processed into a horizontal concatenation of the scaled RGB matrices from 10 frames of RGB images of the resolution 600x1920 taken from the front view of a car.

Output: Whether the car in front, if present, is lane changing to the right, left, or staying in the same lane.

Loss Function: We use a Cross-Entropy Loss function.

$$L_i = -log(\frac{e^{(f_y)_i}}{\Sigma_j e^{f_j}})$$

Dataset

The **Prevention Dataset** [1] is comprised of 5 separate recordings that make up 6 hours and 17 minutes of driving mostly on the freeway. Our work uses the **first recording** (37 mins and 47 km of driving).

- **Collected using:**
  - 6 High-Speed FHD+ mounted to the front of the car
  - Lidar detectors stationed on top of the car
  - Radar detectors all around vehicle

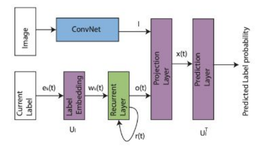
The dataset provides users with multiple files that include lane changing data, lane labeling data, trajectories, and detections.

- **Our data processing pipeline:**
  - Splice 37 min of video into 23845 jpg images (video frames)
  - Run OpencV imread to convert each frame to RGB matrix
  - Scale down to 25% for CNN baseline, scale down to 50% for Encoder-RNN Decoder and Vision Transformer (ViT)
  - Concatenate historical frames (4 or 10) horizontally and run additional feature extraction in case of ViT
  - Split into ⅝ train ⅛ test set (19,871 train frames, 3,974 test frames)

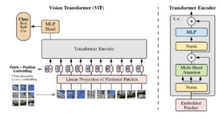
Methods

- CNN Baseline: Similar to most related works, Inspired from An et. al (2020) [2], which achieved state of the art results on MNIST classification. Our architecture: 2D conv layer -> ReLU -> 2D conv layer -> ReLU -> MaxPool2D -> dropout -> flatten -> fc layer -> dropout -> fc layer -> logits

- CNN Encoder-RNN Decoder: Heavily relied on Wang et. al (2016) [3] implementation.
  - CNN Encoder: ResNet152 model pretrained on ImageNet dataset from torchvision models. Replaced last fc layer with linear layer with output embed size of 3.
  - RNN Decoder: Encoder embeddings -> LSTM layer (maintaining hidden state) -> linear layer -> logits



- Pretrained ViT: We finetuned ViT twice, once using 4 historical frames and another time using 10 historical frames, both to predict 10 frames ahead. We modified the number of output classes in the final layer to be 3 for finetuning.
  - Architecture: Patch Normalization -> multiheaded attention -> Merging + Normalization -> Multi-Layer Perceptron -> Softmax -> Logits



Results

The ViT models obtained the greatest testing accuracies, while the baseline CNN model obtained the greatest training accuracy.

Model	CNN Baseline	CNN RNN EncDec	ViT 10 Frames	ViT 4 Frames
Training Accuracy	86.22%	82.13%	84.51%	82.13%
Testing Accuracy	80.87%	12.4%	81.23%	81.1%

Table 1. Models' Training and Testing Accuracies

Analysis and Conclusion/Future Work

To understand the models more, we found the true/false positive/negative ratios. The ratios are defined as the true/false positive/negative numbers divided by the number of true/false examples.

Model	CNN Baseline	CNN RNN EncDec	ViT 10 Frames	ViT 4 Frames
True Positive Ratio	1.71%	100.0%	0.0%	0.0%
True Negative Ratio	98.29%	0.0%	100.0%	100.0 %
False Positive Ratio	2.01%	100.0%	0.0%	0.0%
False Negative Ratio	97.99%	0.0%	100.0%	100.0%

Table 2. Models' True Positive and Negative Ratios and False Positive and Negative Ratios

This table offered the following insights:

1) The ViT models only predicts no LC
2) The CNN Enc-RNN Decoder only predicts LC
3) The CNN baseline was able to predict both (albeit poorly)

Figure 5. Orginal Observation Window

Figure 6. Mapping of most important pixels to the Model

CNN baseline was able to properly focus

1) lane markings
2) the car spaces in front in the same, right, and left lane.

By running ViT-4 and ViT-10, we saw little difference in testing accuracies, so not much more information was present before the 14 frames before LC. We were able to perform on par with Biparva et al, who used a CNN based architecture on a much larger dataset.

Although the ViT obtained high testing accuracies, the CNN baseline could be considered the best model as it was the only model that was able to discriminate and predict both LC and no LC.

Future work: different dataset, loss weighing, different observation windows, classification at different times, different pre-trained ViT + object detection

References

[1] R. Izquierdo, A. Quintanar, I. Parra, D. Fernandez-Llorca, and M. A. Sotelo. The prevention dataset: a novel benchmark for prediction of vehicles intentions. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 3114–3121, Oct 2019.

[2] Sanghyeon An, Minjun Lee, Sanglee Park, Heerin Yang, and Jungmin So. An ensemble of simple convolutional neural network models for mnist digit recognition, 2020

[3] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification, 2016.