Computer Vision–Based Autonomous Driving With PilotNet+LSTM

Zixin Chen
Department of Computer Science
Yale University
New Haven, CT 06511
zixin.chen@yale.edu

Abstract

Deep learning and computer vision are essential for modern autonomous driving. We propose PilotNet+LSTM, a lightweight LSTM-based extension to Nvidia PilotNet for predicting steering angles from sequences of image frames. By combining PilotNet with an LSTM, PilotNet+LSTM is able to leverage sequential frames to avoid one-shot identification errors and attain better accuracy, robustness and generalization abilities in autonomous driving tasks. Furthermore, it can be efficiently trained via initialization of its CNN weights from a pretrained PilotNet CNN, and impose minimal overhead in online evaluations.

1 Introduction and Motivation

Autonomous driving a key technology that may enable safer, more reliable and more efficient transportation. The main task of an autonomous driving model is to navigate a vehicle in real time. Typically, the inputs to the model consist of live video data from the perspective of a dashcam as a continuous sequence of *frames*, and the model must decide the controls to be sent to the vehicle, such as steering, throttle and braking commands. Concretely, the deep learning task of autonomous driving is to devise a model g that performs regression with a length-d sequence of image frames $F_0, F_1, \ldots, F_d$ as input and real-valued control outputs $\{A\}$. Since autonomous driving happens in real time, the model must be able to ingest high-throughput data and repeatedly produce predictions in very short intervals. This requirement limits the complexity of the neural network and calls for a highly-optimized architecture.

The problem is interesting on two levels. From an application perspective, autonomous driving has the potential to become a backbone of future transportation systems. Through this study, we explore the use of deep learning in this emerging domain. More generally, the real-time deep-learning approaches suitable for processing autonomous driving data may find value in other cases like medical imaging.

2 Background and Related Work

One of the first and most impactful autonomous driving model is Nvidia PilotNet [1]. The model utilizes a CNN to map from an RGB single-frame image to a real-valued steering angle output. While the model is capable of predicting the steering angle in a variety of cases, it is prone to *one-shot errors*. For instance, if the camera is obstructed by debris or overexposed in single frames, the network may produce erratic outputs.

Since its inception, some extensions to PilotNet have been proposed [2]. One notable example is AdmiralNet by Weiss et al., where a 3D convolutional network is employed for encoding intra-frame information [2]. However, with increased model complexity comes degradation to training and evaluation performance and heavier demands to the autonomous driving hardware. In this context, a lightweight extension to PilotNet with minimal overhead can be highly desirable.

3 Model Design

3.1 Architecture

As shown in Figure 1, PilotNet+LSTM consists of the following components:

- A CNN, identical to the CNN layers of the PilotNet model. The CNN admits a three-channel 66-by-200 image as input, and produces a 64-channel 1-by-18 embedding as output.
- A sequence of fully-connected layers that maps from 1064 dimensions to 100 dimensions.
- An LSTM of length 5 that maps from 100 dimensions to 100 dimensions.
- A sequence of fully-connected layers that maps from 100 dimensions to the one-dimensional output space.

The CNN and fully-connected layers sequentially form the original PilotNet model. While PilotNet admits a single frame as input, PilotNet+LSTM admits a sequence of five frames and outputs a single steering angle.

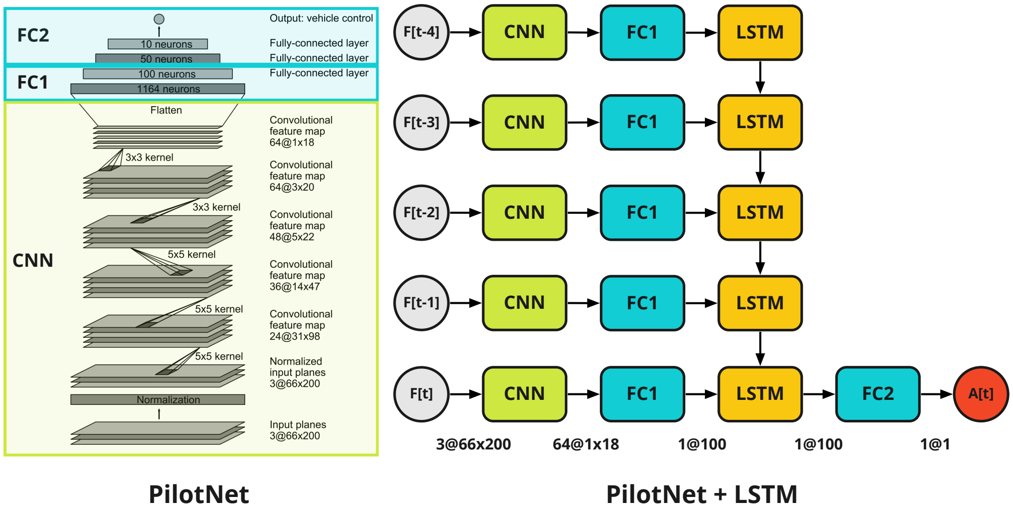


Figure 1: Architecture schematics of the PilotNet and PilotNet+LSTM models.

3.2 Training

Since the CNN layers of PilotNet+LSTM are identical to those of the PilotNet model, the training process of PilotNet+LSTM can be significantly expedited by incorporating pretraining the CNN weights of a PilotNet network. Given a sequence of frame-angle pairs, we first train a PilotNet on single-frame inputs. After convergence, we initialize the CNN weights of a PilotNet+LSTM from the pretrained PilotNet CNN. We show in Section 4.2 that this approach leads to rapid convergence in PilotNet+LSTM training.

4 Evaluation and Results

4.1 Dataset Preparation and Training

We evaluate the performance of PilotNet+LSTM against PilotNet on Sully Chen's real-life driving dataset [3]. The dataset consists of 63,825 RGB image frames of 455-by-256 pixels. Each frame resembles a front-facing view from the perspective of a camera mounted on the dashboard of a car, and is annotated by the angle of the steering wheel (in degrees) at the corresponding time. Together, the frame-angle pairs form a sequence of around 55 minutes of continous driving around Rancho Palos Verdes and San Pedro, California. We preprocess the data as follows:

- Frame-angle pairs corresponding to erroneous angle values are discarded.
- Frames are cropped along the height to remove part of the sky and the vehicle's hood.
- Frames are downsampled to a dimension of 200-by-60.
- Frames are normalized such that the per-channel pixel values are between 0 and 1.

Then, the data is split in the following fashion:

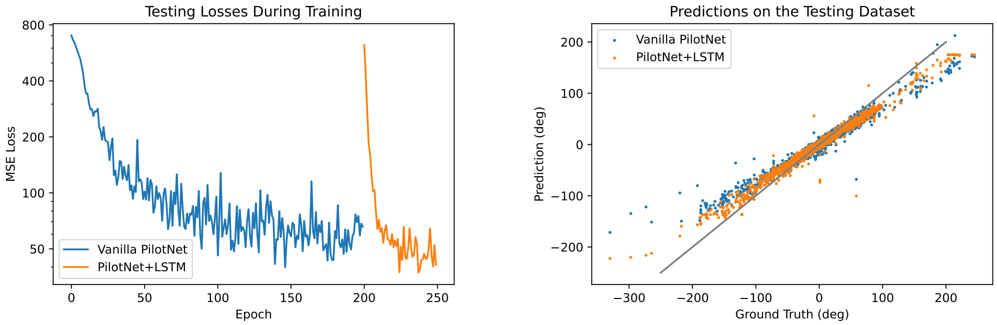
- **Training dataset:** 70% of all data, sampled randomly from the first 90% of the dataset.
- **Testing dataset:** 20% of all data, sampled randomly from the first 90% of the dataset.
- **Generalization dataset:** the last 10% of the dataset, with the original order preserved.

While the training and testing datasets are standard, we utilize the generalization dataset to examine each model's ability to generalize to data that is not close to the training dataset. Since the generalization dataset is taken as the last 10% of all data, there would be no interlacing between it and the training dataset. Therefore, good performance in the generalization dataset would indicate that the models are not merely learning to interpolate between frames in the training dataset.

We train PilotNet for 200 epochs using the Adam optimizer with a learning rate of 10^{-3}. As discussed in Section 3.2, we use the final parameters for the CNN in PilotNet to initialize the PilotNet+LSTM CNN. The parameters in the LSTM and fully-connected layers in PilotNet+LSTM are randomly initialized. We then proceed to train PilotNet+LSTM for 50 epochs. To facilitate regularization, we add a dropout layer with probability 0.5 after all fully-connected layers in each model, except the output layers.

4.2 Prediction Performance

As shown in Figure 2a (note the exaggerated slope due to the log-scaled y-axis), both models successfully converge with no noticeable overfitting. Compared to the vanilla PilotNet model, PilotNet+LSTM with pre-trained CNN weights converge in significantly fewer epochs, and to a lower loss. Table 1 displays the loss values of the models on each of the three datasets. In all cases, PilotNet+LSTM exhibits around a 40% decrease from PilotNet in loss. This result demonstrates that PilotNet+LSTM performs better than PilotNet in the prediction task.



(a) Testing losses per epoch during training. The PilotNet is trained for 200 epochs, and the PilotNet+LSTM is initialized from the CNN weights of PilotNet and trained for another 50 epochs.

(b) Predictions of PilotNet and PilotNet+LSTM on the testing dataset after training. The grey line shows the perfect correspondence between predictions and ground truths. The sign of the angles represents direction.

Figure 2: Losses and predictions on the testing dataset with PilotNet and PilotNet+LSTM models.

This result is further evidenced by a comparison of the PilotNet and PilotNet+LSTM predictions against ground truth in Figure 2b. The predictions from each model exhibit a linear relation with the ground truth, with increased errors when the steering angle is large. Comparatively, the PilotNet+LSTM predictions fall closer to the true steering angles. The error pattern is indicative of the imbalances in the source dataset where steering angles close to zero (i.e., straight-line driving) are predominant. This biases the models toward predicting a smaller steering angle.

Model	Training (70%)	Testing (20%)	Generalization (10%)
PilotNet	63.11	65.85	47.29
PilotNet+LSTM	37.72	40.43	27.16

Table 1: Loss values of PilotNet and PilotNet+LSTM on training, testing and generalization data.

4.3 Generalization Abilities and Robustness

Since the generalization dataset is arranged sequentially, it is instructive to display the prediction performance of each model on the time axis of frames. Figure 3 displays the difference between ground truth and the predicted values for each model across the generalization dataset. Observe that in almost all frames, PilotNet+LSTM performs better than PilotNet with a prediction error closer to zero. In a few regions, PilotNet exhibits large error spikes not present in PilotNet+LSTM predictions. One such region (frames 4900 to 5200) is highlighted with a red rectangle and displayed in detail in Figure 4. At frame 5100 within the region, PilotNet produces a single-frame error spike of value -36.87, significantly higher than the surrounding error values of -9.75 at frame 5099 and -13.32 at frame 5101. This error spike is not present in the PilotNet+LSTM results.

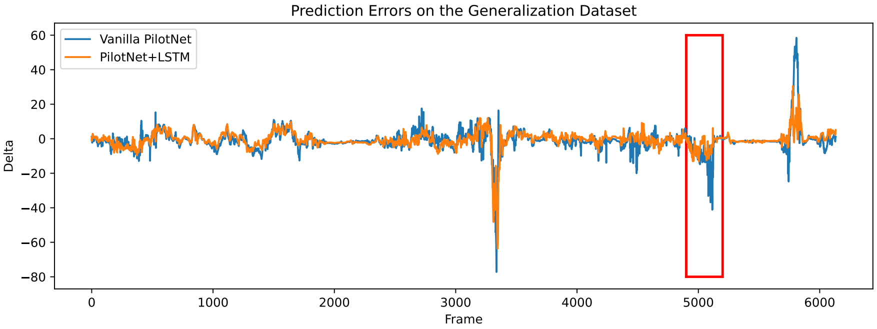


Figure 3: Prediction errors (i.e., deltas between predicted angles and ground truths) on the generalization dataset with PilotNet and PilotNet+LSTM models. The red rectangle marks a region from frame 4900 to frame 5200 where PilotNet+LSTM performs significantly better than PilotNet does.

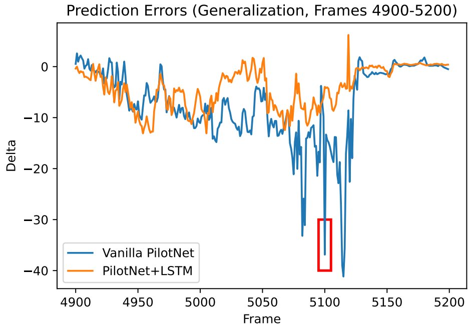


Figure 4: Detailed view of the prediction errors between frames 4900 and 5200 on the generalization dataset. The red rectangle marks a single-frame spike in PilotNet error at frame 5100 that is not present in the PilotNet+LSTM model.

This difference in the behavior of the models can be explained by visualizing the CNN activation maps around frame 5100. A standard approach is to highlight important regions in a convolutional layer by backpropagating the loss gradient, with methods like *Gradient-weighted Class Activation Mapping (Grad-CAM)* [4]. Since Grad-CAM is originally defined for classification problems, we modify it for regression activation mapping (RAM) by substituting the loss function with MSE loss between the predicted angles and the ground truths. Figure 5 display the Grad-RAM activation maps of the second PilotNet convolution layer on frames 5098 through 5102. Observe that throughout this image sequence, the road surface is covered by horizontal stripes of shadows. As a consequence, PilotNet misses the centerline of the road in frame 5100. However, it manages to identify part of the centerline and the vehicle ahead in the rest of the frames. This single-frame misidentification is likely the reason behind the error spike. Since PilotNet+LSTM is capable of encoding the information in the preceding frames via the LSTM hidden states, this issue does not impact the accuracy of the PilotNet+LSTM predictions significantly. This difference highlights the robustness of PilotNet+LSTM against one-shot errors in the CNN.

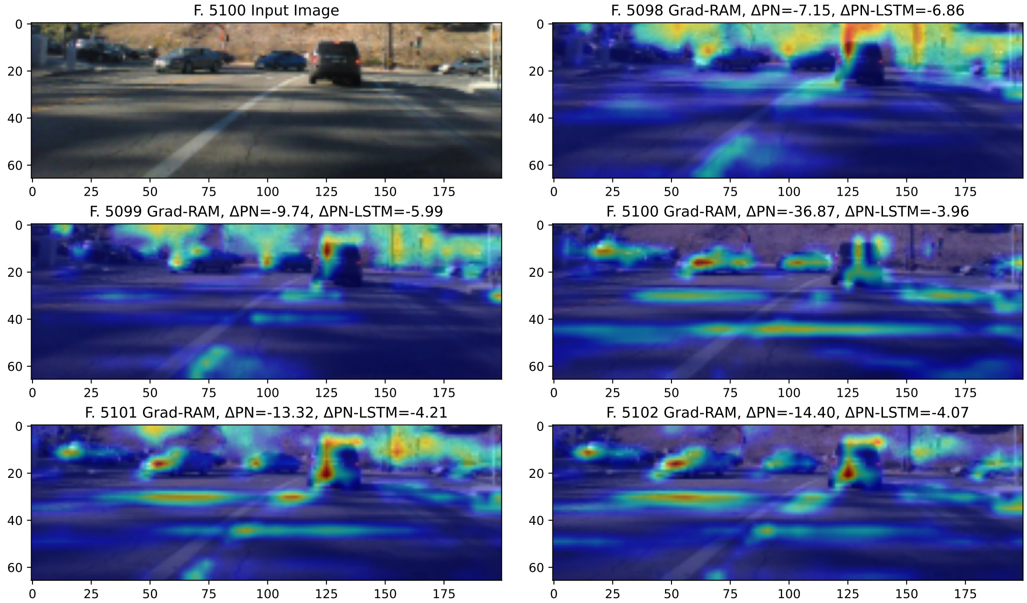

F. 5100 Input Image
F. 5098 Grad-RAM, ΔPN=-7.15, ΔPN-LSTM=-6.86
F. 5099 Grad-RAM, ΔPN=-9.74, ΔPN-LSTM=-5.99
F. 5100 Grad-RAM, ΔPN=-36.87, ΔPN-LSTM=-3.96
F. 5101 Grad-RAM, ΔPN=-13.32, ΔPN-LSTM=-4.21
F. 5102 Grad-RAM, ΔPN=-14.40, ΔPN-LSTM=-4.07
0
20
40
60
0
25
50
75
100
125
150
175

Figure 5: Grad-RAM activation maps of the second PilotNet convolution layer between frames 5098 and 5102, alongside the input at frame 5100. Regions with hues closer to red indicate greater importance for predicting the angle.

5 Conclusions and Future Work

Deep learning and computer vision are vital aspects to modern autonomous driving systems. While CNN-based methods like PilotNet are effective in predicting steering angles from single-frame information, they fail to account for relations between chronologically adjacent frames, and are prone to one-shot errors in individual frames. By combining PilotNet with an LSTM, PilotNet+LSTM is able to leverage sequential frames to attain better accuracy, robustness and generalization abilities in autonomous driving tasks. Furthermore, it can be efficiently trained via initialization of its CNN weights from a pretrained PilotNet CNN, and imposes minimal overhead in online evaluations.

Further Work in this project are concentrated in two key areas: data and model. Data-wise, we aim to develop augmentation and normalization strategies to combat the imbalances in real-world driving datasets. Specifically, we hope to devise methods to enhance the representation of infrequent scenarios (e.g., large steering angles, hard brakes) in driving data. In terms of model, we look to extend the LSTM to account for data sampled at irregular intervals, using techniques like positional encoding or interpolation networks. Together, these objectives may contribute to the development of more performant and robust autonomous driving systems of tomorrow.

References

[1] Bojarski, M. & Chen, C. (2020) The NVIDIA PilotNet Experiments. https://arxiv.org/pdf/2010.08776.pdf

[2] Weiss, T. & Behl, M. (2020) DeepRacing: A Framework for Autonomous Racing. https://www.madhurbehl.com/newpubs/weiss2020deepracing.pdf

[3] Sully, C. (2020) driving-datasets. https://github.com/SullyChen/driving-datasets

[4] Selvaraju, R. R. & Cogswell, M. (2019) Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. https://arxiv.org/pdf/1610.02391.pdf