End-to-end Lane Shape Prediction with Transformers

DSE-316/616, Deep Learning

Course Instructor: Dr. Vinod Kurmi

Sharanyak Podder (19287)

Saswata Sarkar (19279)

IISER Bhopal

Submission date: 24 November 2022, 11:55 pm

Contents

1 Introduction

In recent years, autonomous driving has advanced quickly, attracting the full focus of both academia and business. The "eyes" of autonomous driving, the perception task, heavily relies on lane detection to comprehend the road environment. It has to do with how autonomous vehicles position themselves, obey traffic laws, and then decide how to drive.

Lane detection is difficult for two key reasons. First off, the lanes have a narrow shape, a single structure, and uncommon visual cues. Furthermore, lanes may disappear as a result of numerous factors such wear and tear, shadow occlusion, traffic congestion, poor weather, or glaring light. However, using information like as vehicle alignment, road layout, and the visibility of nearby lane markings, humans can locate lane lines with ease. People discovered that the global information incorporating extra visual cues is preferable to identifying the lane lines since it is inspired by the human visual system.

Existing approaches [[1][2][3]] significantly outperform conventional approaches that are based on manually created features and the Hough Transform in terms of lane detection tasks by utilising the powerful representation capabilities of convolution neural networks (CNNs). However, current CNNs-based approaches to tackling the aforementioned issues are still insufficient. The earlier techniques [4] frequently generate segmentation results first, followed by post-processing techniques like segment clustering and curve fitting. When learning to segment lanes, these methods are ineffective and disregard global context. Some approaches to the context learning problem involve message passing or additional scene annotations to capture the overall context for improving ultimate performance, but these approaches inevitably take longer and require more data. A soft attention-based strategy, as opposed to these ones, creates a spatial weighting map that distils a richer context without external consumes. While considering the relationships between features that enable to infer slender structures, the weighting map can only measure the feature's relevance.

We propose trying to remake the lane detection output as parameters of a lane shape model in order to address these problems. We also suggest creating a network made of non-local building blocks to strengthen the learning of global context and lane slender structures. The result for each lane is a set of parameters that, using an explicit mathematical formula derived from road structures and the camera pose, roughly replicate the lane marker. Without using any 3D sensors, such metrics can be used to determine the road curvature and camera pitch angle given specific priors like camera intrinsic. The next step is to draw inspi-

ration from natural language processing models, which frequently use transformer blocks [5] to explicitly express long-range dependencies in language sequence. We build a transformer-based network that synthesises information from any pairwise visual features, allowing it to capture the long, thin structures, and global context of lanes. When trained end-to-end with a Hungarian loss, the entire architecture simultaneously predicts the suggested outputs.

The classic "TuSimple" lane detection benchmark is used to verify the effectiveness of the proposed technique. With the smallest model size and the fastest speed, our technique achieves state-of-the-art accuracy while having the lowest false positive rate.

2 Method

Our end-to-end method reframes the output as parameters of a lane shape model. Parameters are predicted by using a transformer-based network trained with a Hungarian fitting loss.

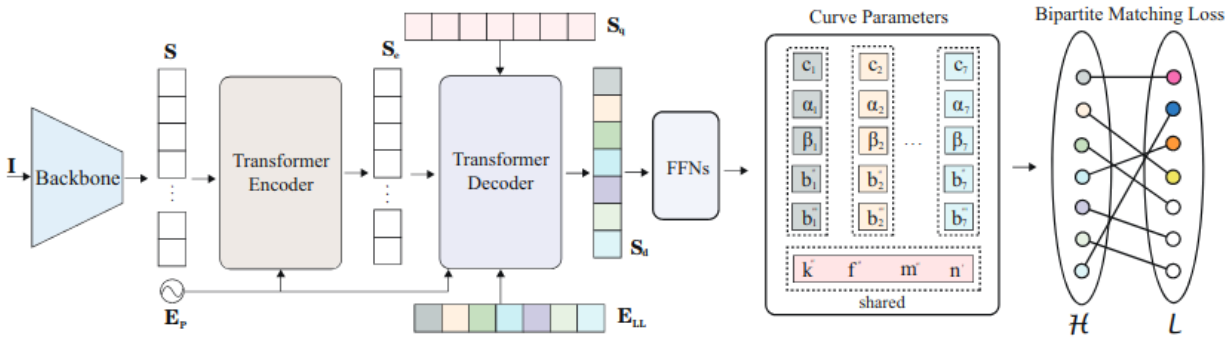

Figure 1: Model Architecture

2.1 Lane Shape Model

The prior model of the lane shape is defined as a polynomial on the road. Typically, a cubic curve is used to approximate a single lane line on flat ground:

$$X = kZ^3 + mZ^2 + nZ + b, \tag{1}$$

Here, k, m, n, b are real number parameters, $k \neq 0$. The (X, Z) represents the point on the ground plane. The curve projected from the road onto the image plane is:

$$u = k'/v^2 + m'/v + n' + b' \times v, \tag{2}$$

Here, k', m', n', b' are composites of parameters and camera intrinsic and extrinsic parameters. The (u, v) is a pixel at the image plane.

In the case of a tilted camera whose optical axis is at an angle of ϕ to the ground plane, the curve transformed from the untitled image plane to the tilted image plane is:

$$u' = k' \times \cos^2\phi/(v' - f\sin^2\phi)^2 + m' \times \cos\phi/(v' - f\sin\phi) + n' + b' \times v'/\cos\phi - b' \times f\tan\phi, \quad (3)$$

Here, f is the focal length in pixels, and (u', v') is the corresponding pitch-transformed position.

Curve re-parameterization. By combining parameters with the pitch angle ϕ, the curve in a tilted camera plane has the form of:

$$u' = k''/(v' - f'')^2 + m''/(v' - f'') + n' + b'' \times v' - b''' \quad (4)$$

2.2 Hungarian Fitting Loss

In order to identify positives and negatives, the Hungarian fitting loss conducts a bipartite matching between projected parameters and ground truth lanes. The Hungarian method effectively resolves the matching problem. Then, lane-specific regression losses are optimised using the matching result.

Bipartite matching. The method predicts a fixed N curves, where N is set to be larger than the maximum number of lanes in the image of a typical dataset. Let us denote the predicted curves by

$$\mathcal{H} = \{h_i | h_i = (c_i, g_i)\}_{i=1}^{N},$$

Where, $c_i \, \epsilon \, \{0, 1\}$ $(0 : non - lane, 1 : lane)$. Since the number of predicted curves N is larger than the number of ground truth lanes, we consider the ground truth lanes also as a set of size N padded with non-lanes

$$\mathcal{L} = \left\{\hat{l}_i | \hat{l}_i = (\hat{c}_i, \hat{\mathbf{s}}_i)\right\}_{i=1}^{N}.$$

We formulate the bipartite matching between the set of curves and the set of ground truth lane markings as a cost minimization problem by searching an optimal injective function $z : L \to H$, i.e., $z(i)$ is the index of curve assigned to fitting ground-truth lane i:

$$\hat{z} = \arg\min_{z} \sum_{i=1}^{N} d\left(\hat{l}_i, h_{z(i)}\right),$$

2.3 Architecture

The architecture shown in Fig. 1 consists of a backbone, a reduced transformer network, several feed-forward net- works (FFNs) for parameter predictions, and the Hungarian Loss. Given an input image I, the backbone extracts a low-resolution feature then flattens it into a sequence S by collapsing the spatial dimensions. The S and positional embedding E_p are fed into the transformer encoder to output a representation sequence S_e. Next, the decoder generates an output sequence S_d by first attending to an initial query sequence S_q and a learned positional embedding E_{LL} that implicitly learns the positional differences, then computing interactions with S_e and E_p to attend to related features. Finally, several FFNs directly predict the parameters of proposed outputs.

3 Paper's result

Datasets. The widely-used "TuSimple" lane detection dataset is used to evaluate our method. The TuSimple dataset consists of 6408 annotated images which are the last frames of video clips recorded by a high-resolution (720×1280) forward view camera across various traffic and weather conditions on America's highways in the daytime. It is split initially into a training set (3268), a validation set (358), and a testing set (2782) .

Method	FPS	MACs	Para	PP	Acc	FP	FN
PolyLaneNet	115	1.784	4.05	-	93.36	.0942	.0933
Paper's result	420	0.574	0.77	-	96.18	.0291	.0338

In this section, they treat PolyLaneNet [6] as the baseline method since they also predict parametric output for lanes and provide amazingly reproducible codes and baseline models. The proposed method was trained using both TuSimple training and validation set as previous works did. The time unit compares the FPS performance, and they also report MACs and the total number of parameters. All results are tested on a single GTX 1080Ti on their platform.

4 Our reproduced result

We have reproduced the results of this paper[7] and use that paper as our base paper and base result.

Method	FPS	MACs	Para	PP	Acc	FP	FN
Our result	143.066	-	-	-	95.94	0.0312	0.0362
Our improved result	135.27	-	-	-	96.08	0.0314	0.337

We have run the base model at two occasions and found two results, second one being effective, with a 96.08%.

5 Additional Experiment

We have tested the model with some manually collected images, which are being collected in IISERB campus in daylight with a normal phone camera. While we tested the images, we got encountered with the "Horizon Issues"- the predicted lines are going beyond the horizon.

To resolve this issue, we have taken a pre-trained segmented model(UNet), trained on CityScapes dataset. We passed normal images and segmented images to the trained model to test the results. Now, It is giving significant result.

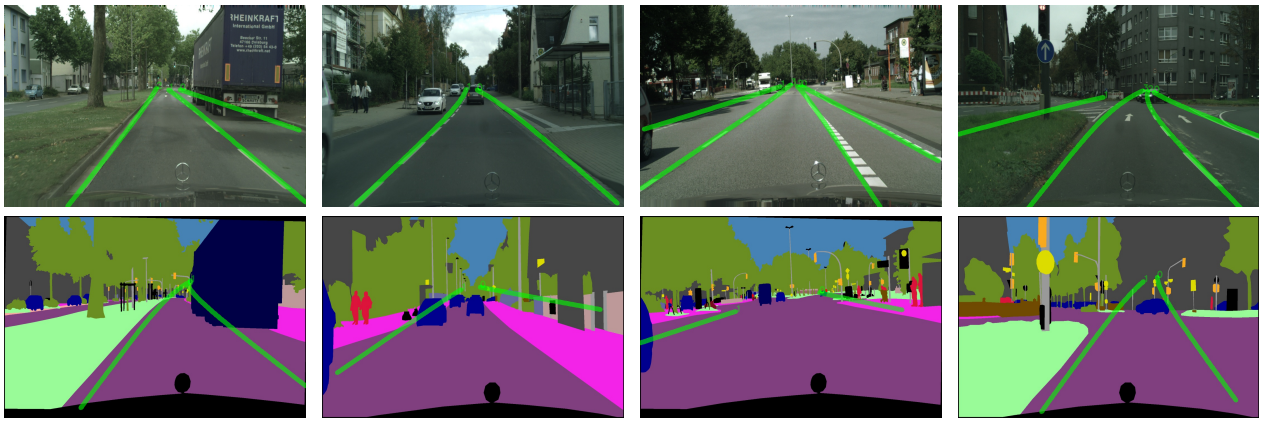

Figure 2: Normal and Segmented Images

6 Conclusion

From the experiments we have done, we understood that the model in our base paper has some lack in robustness. the model can predict lane shapes perfectly when the input images are taken on the road by moving vehicle. It fails to correctly produce the expected results if the input image is taken form any of the side of the road. We think there is room for the development of this nature of the model and the end-to-end model with transformers can become more robust to predict the Lane-Shape.

7 Contributions

7.1 Sharanyak:

Additional experiment done with segmented images.

7.2 Saswata:

Reproduced the base paper results.

8 Script

8.1 LSTR:

```
Login to the remote server with the command ssh dl@172.30.1.163
Password : dl@2022
Go to the folder with the command

cd /data3/dl/grp11/TuSimple/LSTR/

To reproduce the result, run the following command:

python test.py LSTR --testiter 500000 --modality eval --split testing --
    batch 16

To visualizze the test images run the following command

python test.py LSTR --testiter 500000 --modality eval --split testing --
    debug

Go to the folder to see the test images with the command

cd /data3/dl/grp11/TuSimple/LSTR/results/LSTR/500000/testing/lane_debug


Store cutom images in the ./images folder and run the following command
    save the test images in the ./detection folder

python test.py LSTR --testiter 500000 --modality images --image_root ./ --
    debug
```

References

[1] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018.

[2] Jonah Philion. Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11582–11591, 2019.

[3] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] Jianwei Niu, Jie Lu, Mingliang Xu, Pei Lv, and Xiaoke Zhao. Robust lane detection using two-stage feature extraction with curve fitting. *Pattern Recognition*, 59:225–233, 2016.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[6] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021.

[7] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021.