CSE 565 Computer Vision

HW 4 PAPER PRESENTATION REPORT

Generating Dynamic Kernels via Transformers for Lane Detection *"October 2023"*

[2023 IEEE/CVF International Conference on Computer Vision (ICCV)]

Ercan AKCAN 244201001033

Contents

Introduction

Lane detection is a fundamental task in *Autonomous Driving System (ADS)* and *Advanced Driver Assistance System (ADAS)* (*for example driving route planning, lane keeping assist, and adaptive cruise control etc.*). Lane detection in real scenes is very challenging.

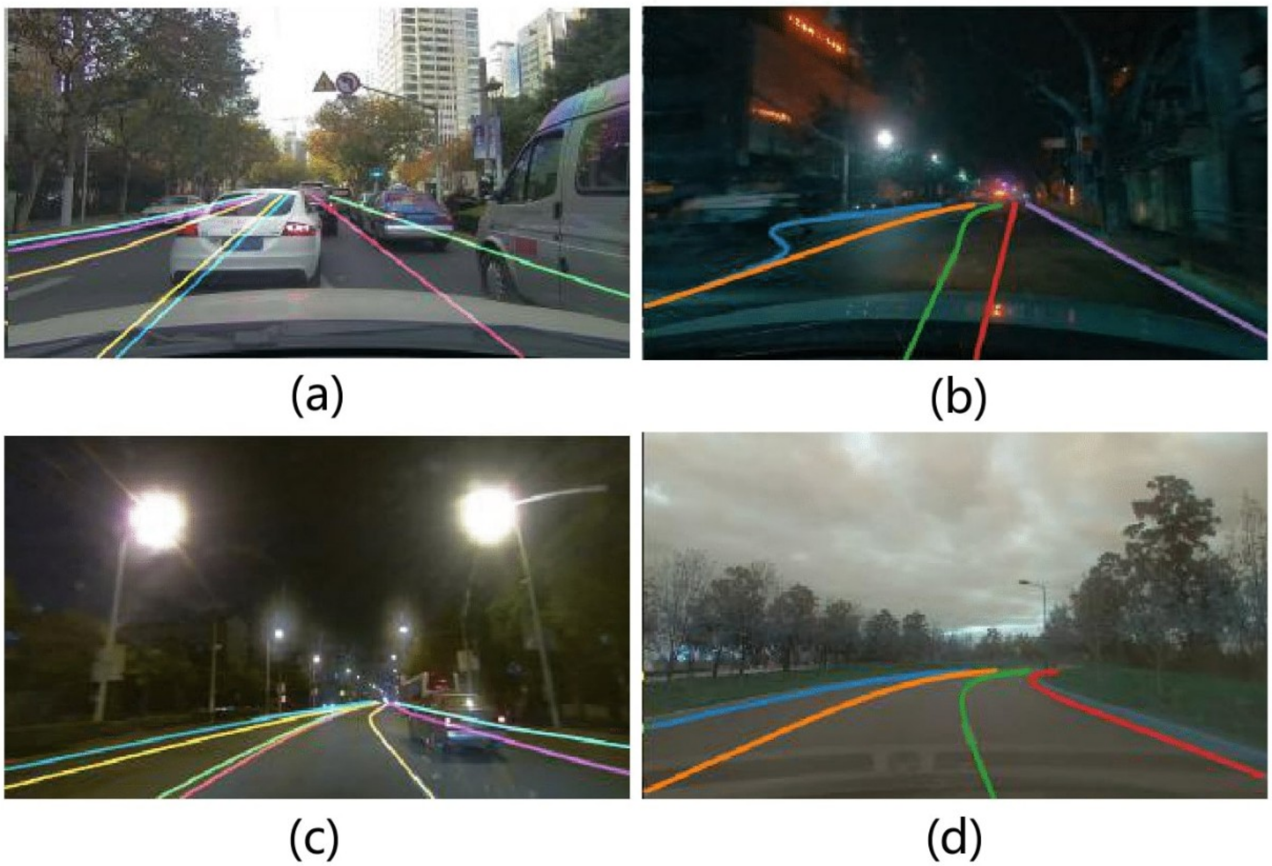


Figure 1: Examples of lanes lines in real scenes. The lane lines often have complex topologies, additionally the lighting and weather conditions in these four pictures are also very different.
(a) The lane lines are blocked by vehicles.
(b) The lane lines have forked and curved structures.
(c) The lane lines are dense and blocked.
(d)The lane lines are extremely curved.

Traditional Lane Detection Methods

Traditional lane detection methods usually rely on hand-crafted features and post-processing techniques like ***Hough Transform***. *[2012, 2014, 2015, 2016]* These methods are limited in representation ability and robustness, making them difficult to handle the diversity of lane lines in different road scenarios.

Recent Lane Detection Methods

Deep-learning-based lane detection methods have recently achieved great success, thanks to the powerful representation ability of *convolutional neural networks* ***(CNN)***. Existing methods often rely on specific background knowledge – *such as straight line anchors and parametric curves* – to detect lane lines.

Recently, ***dynamic convolution-based methods***, e.g., *CondLaneNet*, have shown promising performance by considering the features from some key locations of a lane line, – *such as the starting point, as convolutional kernels, and convolution them with the whole feature map* – to detect lane lines.

However, the kernels computed from the key locations fail to capture the lane line's global information due to its *long* and *thin structure*, leading to inaccurate detection of lane lines with complex topologies. In addition, the kernels resulting from the key locations are sensitive to occlusions and lane intersections. For instance, ***when they are obscured by vehicles or pedestrians.*** Additionally, it is difficult to differentiate ***between multiple lane lines that share the same starting point.***

Related Works - Literature review

Anchor-based methods

The anchor-based methods *[LaneATT:2021, SGNet:2021, CLRNet:2022]* detect lane lines by generating a set of pre-defined line anchors first, and then extracting the anchor features with pooling operation, and finally regressing the offsets from the anchors to the target lane lines with the extracted features.

For these methods, the Non-maximum Suppression (NMS) post-processing is needed to remove duplicate predictions, which decreases the efficiency. Also, the fixed lane anchors can hardly deal with the lane lines with complex topologies, such as the curved and forked lane lines, which influences the lane detection performance.

Parameter-based methods

Parameter-based methods *[PolyLaneNet:2021, LSTR:2021]* represent lane lines with parametric curves and detect lane lines by predicting the corresponding curve parameters. It removes the complex post-processing procedures, such as pixel clustering and non-maximum suppression, thus can achieve fast speed.

The prediction for many orders of the parametric curves is difficult and not robust, since a small prediction error on these terms will cause a large change in the shapes of lane lines. Therefore, the parameter-based methods have not exceeded other lane detection methods in accuracy.

Segmentation-based methods

Segmentation-based methods *[LaneNet:2018, FOLOLane:2021, HDMapNet:2022]* detect lane lines with pixel-wise classification. Some of them apply semantic segmentation to detect lane points, which are then processed further to extract lane lines.

This method may fail when the starting points are blocked by vehicles and pedestrians, or shared by multiple lane lines. And the kernels generated from starting points lacking the lane line's global information, making this method not robust for handling lane lines with complex topologies.

The Main Method

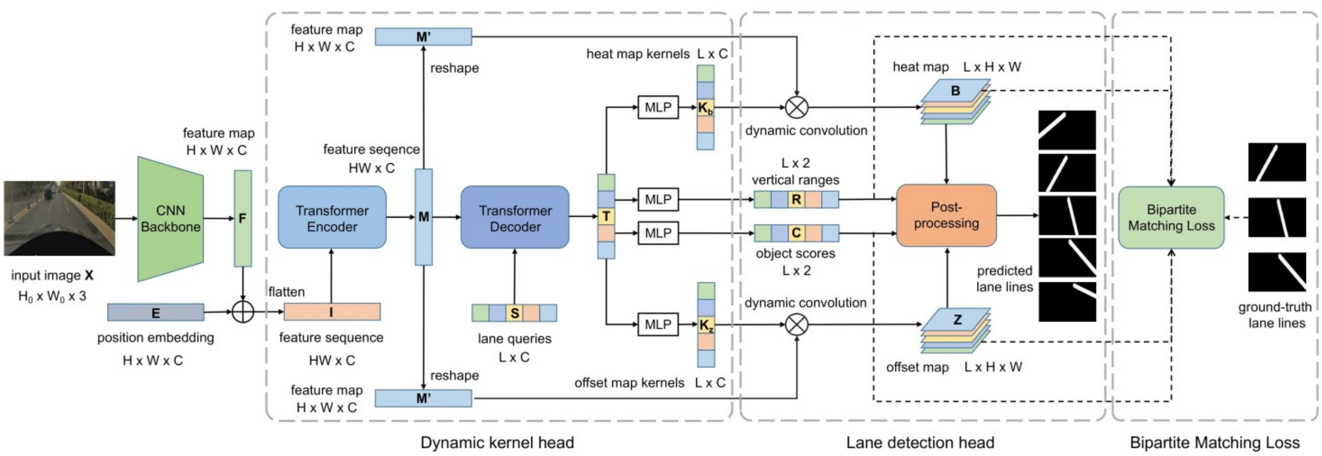
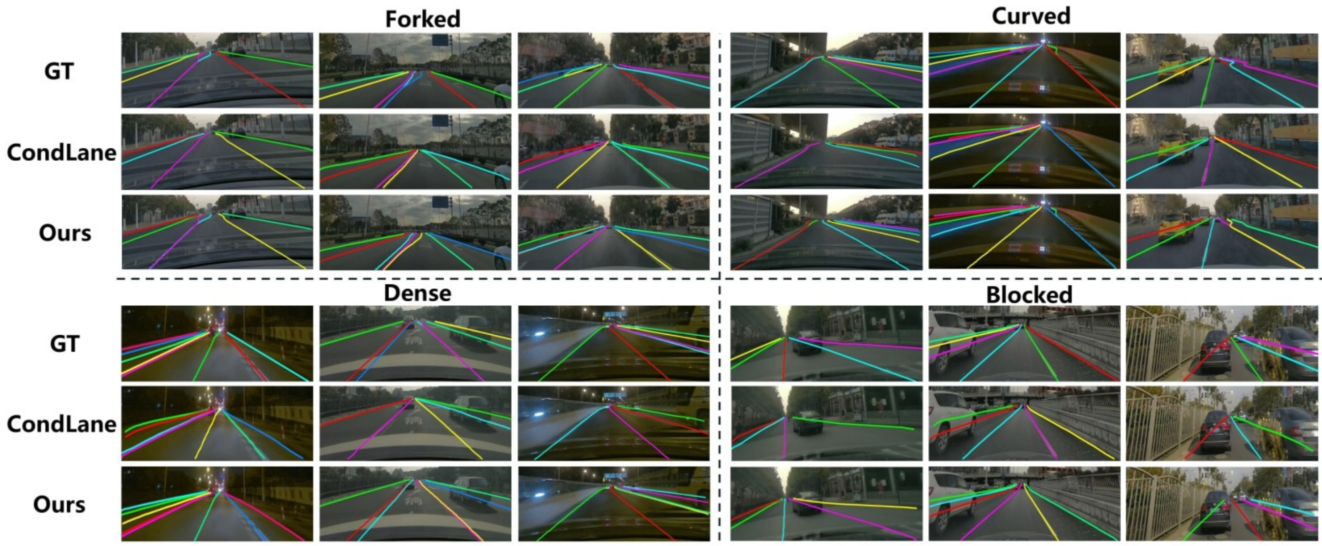


Figure 2: Overall Framework. It includes a CNN backbone to extract a feature map from the input image
Dynamic kernel head *to generate the dynamic convolutional kernels for lane lines via a transformer,*
Lane detection head *to detect lane lines with dynamic convolution,*
Bipartite matching loss *for model training. The dotted arrow parts are only engaged in training.*

The Experimental Results & Discussion

They conduct experiments on four representative lane detection benchmarks: **OpenLane** {160K- 40K}, **CurveLanes** {100K-20K-30K}, **CULane** {89K-1K-35K} and **Tusimple** {3K-.3K-2K} images for training, validation, and testing, respectively, approximately they have **390K** training image. They adopt F1 score as the metric for OpenLane, CurveLanes and CULane datasets.



Method	Backbone	F1 (%)	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Curve	Cross	Night	FPS	GFlops
LaneATT [24]	ResNet-18	75.13	91.17	72.71	65.82	68.03	49.13	87.82	63.75	**1020**	68.58	153	**9.3**
LaneATT [24]	ResNet-34	76.68	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72	129	18.0
LaneATT [24]	ResNet-122	77.02	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81	20	70.5
FOLOLane [22]	ERFNet	78.80	92.70	77.80	75.20	79.30	52.10	89.00	69.40	1569	74.50	40	-
CondLaneNet [13]	ResNet-18	78.14	92.87	75.79	70.72	80.01	52.39	89.37	72.40	1364	73.23	**173**	10.2
CondLaneNet [13]	ResNet-34	78.74	93.38	77.14	71.17	79.93	51.85	89.89	73.88	1387	73.92	128	19.6
CondLaneNet [13]	ResNet-101	79.48	93.47	77.44	70.93	80.91	54.13	90.16	75.21	1201	74.80	47	44.8
GANet [28]	ResNet-18	78.79	93.24	77.16	71.24	77.88	53.59	89.62	75.92	1240	72.75	153	-
GANet [28]	ResNet-34	79.39	93.73	77.92	71.64	79.49	52.63	90.37	76.32	1368	73.67	127	-
GANet [28]	ResNet-101	79.63	93.67	78.66	71.82	78.32	53.38	89.86	**77.37**	1352	73.85	63	-
Ours	ResNet-18	80.36	94.11	79.17	73.55	80.39	54.41	90.37	75.89	1214	75.39	105	13.7
Ours	ResNet-34	80.55	94.12	79.72	**77.02**	**82.51**	53.76	90.59	76.65	1370	**75.57**	91	23.2
Ours	ResNet-101	**80.77**	**94.17**	**79.90**	75.43	80.99	**55.00**	**90.97**	76.87	1047	75.11	45	50.2

They propose an architecture which utilizes a transformer to generate dynamic convolutional kernels for each lane line in the input image, and detects lane lines with dynamic convolution.

It can capture the lane line's global information, enabling them to effectively handle occlusions and lane lines with complex topologies. The proposed method is validated on four benchmarks of lane detection, i.e., *OpenLane, CurveLanes, CULane and Tusimple*

The experimental results shows the state-of-the-art performance of the proposed method.

Improvement

- ✔ As I can see, this paper suggests lane detection for 2D images, where is discussed in "*Dynamic Kernel Head*" section. Maybe if we add some addition, i.e., add 3D camera, additionally we will make 3D lane detector.
- ✔ In this paper, a single-layer personal attention mechanism is used. If, instead of singular self-attention, we use multilayer attention, where we use based on weight and height attention in two branches, and then combine them in a final branch, accuracy and performance will better.
- ✔ In section "*Dynamic Kernel Head*", 4 queries were selected from total 80 lane queries. So if we select more of 4 queries, like 8 or 16, maybe accuracy and performance will better.