

Larger Kernel, Attention and Data Matters: Make Curve-based Lane Detection Great again

Chenhui Gou
u7194588

Yunxiang Liu
u7191378

Zicheng Duan
u7170273

Abstract

In this paper, we rethink the curve modeling Lane Detection method. Compared with previous anchor-based and segmentation-based methods, curve based model naturally learns holistic lane representations. Yet the performance of the current curved-based model is lower than the state-of-the-art segmentation-based method, we find two keys that influence the performance of the curved-based model which are global information understanding and preventing overfitting. Thus, We propose three schemes to improve the effectiveness of the curved-based model. Firstly, we introduce a novel Large Kernel Enhanced Attention module, which owns the global modeling ability and has a long shape bias suitable for lane detection. Secondly, we propose a Random Lane Masking augmentation that can mimic difficult detection cases in real-world which induced by occlusion and natural wear. This augmentation can alleviate the overfitting problem of lane detection methods. Lastly, in order to allow this method to be developed into edge computing devices, we implement our Large Kernel Convolution as Depthwise Convolution. Combining these three contributions, our method achieves state-of-the-art performance in existing curved-based methods and achieves SOTA performance and model-size trade-off compared with widely used Semantic segmentation-based methods. Our code can be found in https://github.com/ZichengDuan/ENGN8501_LaneDet.

1. Introduction

Lane detection aims at perceiving the precise position of different lanes on the road scene, which is an elementary task for modern autonomous driving cars. Accurately detecting lanes helps car route planning, lane keeping, and lane changing. Traditional lane detection methods use some hand-craft features to detect lines. However, the performance of these traditional methods is lower compared with the latest deep learning-based methods and may fail when facing complex conditions such as occlusion and low lighting.

The current deep learning methods can be divided into four different categories, namely semantic segmentation-

based, instance segmentation-based, anchor-based, and curve-based methods. The semantic segmentation-based methods [18], [23], [35], [10], [17] pre-define each lane as an individual class and attempt to label each pixel accordingly. Some other popular methods [14, 16, 20], i.e., instance segmentation methods, first identify the foreground pixels and the background pixels. Next, these instance segmentation-based methods will cluster lane pixels into different classes using a variety of clustering techniques. Moreover, inspired by the anchor-based object detection [22], some methods [9, 24, 29, 31] conduct lane detection based on the pre-defined lane anchors. More recently, a few methods directly model line as different parametric curves such as polynomial curve [25] and bézier curve [7] and trains a model to predict the parameters that define the lane.

For current lane detection, the segmentation-based methods and anchor-based methods still dominate this field. However, segmentation methods suffer from heavy computation costs and do not take advantage of the prior knowledge of the lane line, i.e., the geometric shape of the curve line. Lacking a holistic understanding, it is handy for segmentation-based methods to deal with challenging conditions such as occlusion and wear. Although anchor-based methods have prior knowledge of lane line, they need to design anchors which usually varies on different datasets. Such anchor designing is inconvenient for the real-world autonomous driving system. Also, anchor-based methods require post-processing, such as Non-Maximum Suppression (NMS) used in [24], which also limits their adaption ability to different scenes. We assume modeling the lane lines as holistic curves enables the system to have a global understanding of the lane line and would be a preferable representation of a line. Instead of outputting a large segmentation result, curve-based methods only predict a few parameters which allow potentially low computation complexity. However, the performance of the previous curve-based method is still lower compared with other state-of-the-art methods. In a very recent work [7], the authors argue that the lower performances in curved-based methods mainly result from the learning difficulties of the abstract polynomial coefficients. To learn the parameters in an easier approach, the authors model the curve as a cubic

Bézier curve and build a model called BézierLaneNet that is trained end-to-end and is easier to be optimized. In BézierLaneNet, two novel designs are introduced, one is the design of calculating sampling loss instead of calculating loss directly on Bézier control points, the other one is a feature flip fusion module. These two designed modules improve the understandability of global information on car lanes. Furthermore, the experiments in [7] show the performance of curve based method is heavily influenced by the data augmentation due to overfitting problems. Thus we suppose the keys to the curve-based method are global information understanding as well as alleviating the overfitting. To improve the global information perception ability, we propose a Large Kernel Enhanced Attention Block (LKEA) inspired by the recent success of larger kernel convolution [5] and the self-attention mechanism [6]. Through comprehensive experiments, our LKEA module improves the detection performance effectively while increasing only a small amount of parameters. Also, we propose a novel random Line Masking augmentation method that can alleviate the overfitting problem due to the limited dataset size, specifically, we mask the images with either horizontal or vertical lines to mimic different types of broken lanes induced by natural wear and car occlusion.

Finally, we summarize our main contributions as following aspects:

- A novel Large Kernel Enhanced Attention module is proposed, which captures the global information using the attention mechanism and has a long shape bias obtained with large kernel convolution.
- A depth-wise large kernel convolution is used in LKEA module to achieve better performance and parameters trade-off, which allows this model to be easily deployed to some autonomous driving platforms with small memory requirements.
- A random line masking module is designed to conduct augmentation on input images, which mimics the lane break in real-world environments. Random line masking promotes the ability of the model to tackle the overfitting problems on limited datasets.

2. Related Work

Semantic segmentation based method. With the success of the convolution neural network (CNN) in semantic segmentation, several approaches [10, 17–19, 23, 35] regards the lane detection as semantic segmentation problem through tackling each lane as an individual class. [35] combines CNN and Recurrent neural network (RNN) to build a hybrid model which takes in multiple continuous frames to introduce temporal information and can handle some difficult cases, such as vehicle occlusion and unclear lane marks. [10] proposes a

novel EL-GAN pipeline and achieves better results, which learns an embedding between the ground-truth label and the predicted semantic segmentation result. However, semantic segmentation-based methods are normally computationally expensive and will not generalize when lanes number vary. **Instance segmentation based method.** Some methods [16, 20] cast the lane detection problem as an instance segmentation problem. [16] design a two branches network that can first predict the binary segmentation result and pixel embedding result. Then [16] can cluster the foreground pixels into different lane instances. An end-to-end approach is proposed by [20], which can simultaneously predict the lane boundary instance segmentation and classification result. PINet [14] adopts a novel pipeline that first detects the points of road marks and then clusters them into different lane instances. And PINet can be clipped to smaller model sizes due to introducing the Distillation loss. These instance segmentation-based methods normally involve a complex pipeline and lack efficiency because of the requirement of post-processing.

Anchor-based method. Recently, some anchor-based detection methods [9, 24, 29, 31] are developed in lane detection. [9] adopts an anchor per column representation with an intra-network inverse-perspective mapping (IPM), which can handle both image-only and 3D lane detection. LaneATT [24] defines slender lane anchor and proposes novel anchor-based attention to aggregate the global information, which is a one-stage detector and achieves a good speed-accuracy trade-off. Both [29, 31] define anchors as vertical lines. However, anchors need to be carefully designed in these methods.

Curved-based Lane detection. Most recently, several methods [7, 25, 26] directly models the lane line as parametric curves. [26] first treats lane lines as polynomial curves and proposes a differentiable least squares fitting module to fit polynomial curves. [26] can be trained end-to-end to get a weight map that provides accurate line parameters. The PolyLaneNet [25] directly outputs the polynomials belonging to each lane with a simple convolutional neural network, which achieves competitive results and high inference speed.

Attention mechanism in vision. Attention mechanism has been widely developed in computer vision [8, 12, 13, 28, 30, 32]. SE block [13] applies attention along channel dimension and enhances the network representation ability. Taking advantage of attention-module, [12] models object relation and improves objection detection performance. [28] a non-local operation is proposed by to [28] capture the long-range dependencies and achieves competitive results on the video classification task. And in recent years, self-attention is a special kind of attention mechanism, which has been vigorously researched [8, 27, 32].

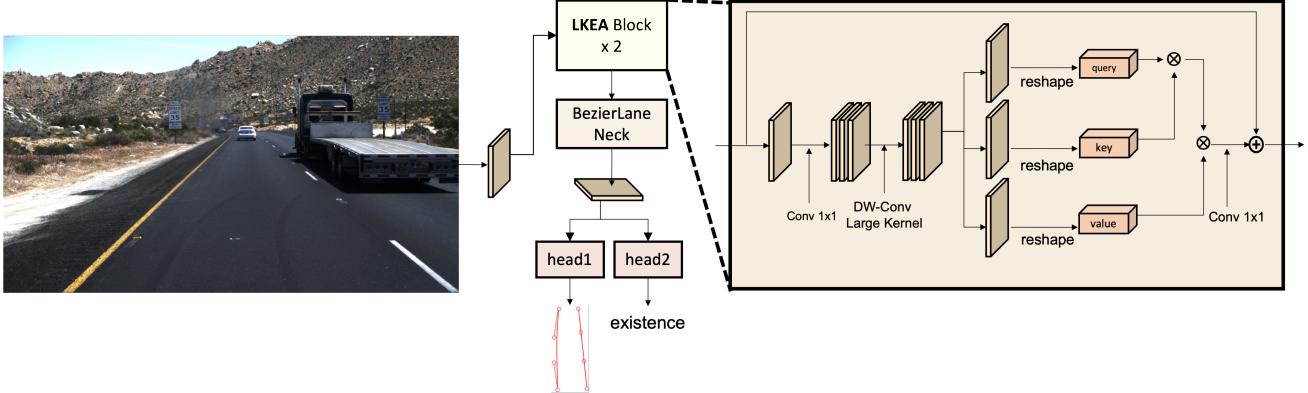


Figure 1. The structure of the proposed network. We concatenate two Large Kernel Enhanced Attention (LKEA) blocks into the origin BézierLaneNet structure. The detailed design of the LKEA is illustrated on the right.

3. Methodology

3.1. Preliminaries on Bézier curves

The Bézier curve is a parametric curve defined by $n + 1$ control points, and its formulation $\mathcal{B}(t)$ is shown in Equation (1):

$$\mathcal{B}(t) = \sum_{i=0}^n b_{i,n}(t) \mathcal{P}_i, \quad 0 \leq t \leq 1, \quad (1)$$

where \mathcal{P}_i is the i -th control point, $b_{i,n}$ are Bernstein basis polynomials of degree n :

$$b_{i,n} = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n. \quad (2)$$

where $\binom{n}{i}$ is a binomial coefficient. Following the previous work [7], we use the cubic Bézier curves ($n=3$) that is verified sufficiently to fit the lane line.

3.2. Large Kernel Enhanced Attention Module

Figure 1 illustrates the overall architecture of our method. Our model is based on the latest BézierLaneNet [7]. In the original BézierLaneNet [7], the model takes in an input image and then uses a backbone such as Resnet18 [1] to extract features. Then [7] designs several blocks to improve the deep features, denoted as BézierLane Neck. These features will then be used to predict several line proposals, also the existence of these line proposals through two separate heads. Each line proposal is represented as eight parameters (2D location of 4 control points). With these control points, lane lines can be recovered through Equation (1).

Compared with naive BézierLaneNet, we add two Large Kernel Enhanced Attention Block blocks before the BézierLane Neck. In later experiments, we found these block effectively improves the performance with only small extra parameters. We design this block inspired by the recent success of Large Kernel Convolution and the self-attention

mechanism. To model the lane line holistically, we observe lane line has some specific properties. The first one is lanes usually have thin and long geometric shapes. The second one is only a few pixels belong to this line although a lane is usually across a large area. Self Attention mechanism allows each pixel to extract information from all other pixels globally. Through attention computation, a query can acquire global information according to the attention score. For example, for a query on a certain line, this query can locate other areas which belong to the same line by finding a high attention score area. A relatively lower attention score indicates this area may belong to other lines. And a very low attention score represents the background area. Thus we propose Self Attention is suitable for lane detection tasks.

Previous transformer-based methods heavily use attention blocks and are difficult to optimize because of losing spatial information when doing attention operations [5]. This precise spatial information is necessary for locating a line. Recent research shows that Large kernel convolution helps transformers learn both the relative position and the absolute position [5]. Thus we use Large kernel convolution before our attention block. Also, the performance gaining brought by this block depends on the kernel size, which is illustrated in the ablation study part. We hypothesize this phenomenon occurs because the large kernel can learn long shape bias, which is also an observation provided in previous larger kernel paper [5]. Moreover, to further decrease the parameters, we use 1×1 convolution followed by a depth-wise large kernel convolution instead of directly applying large kernel convolution. The LKEA structure is illustrated in Figure Figure 1. Through such design, our LKEA module brings noticeable improvement with small parameters increment.

3.3. Random Lane Masking augmentation

We design a random line masking augmentation to simulate comprehensive real-world driving conditions. Existing

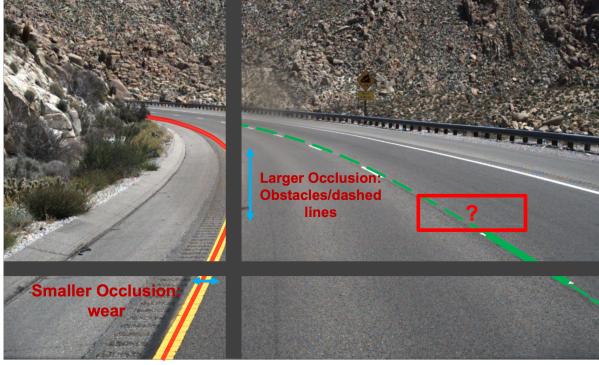


Figure 2. An illustration of our random line masking. We mask the images with either horizontal or vertical masks, where the horizontal masks aim at simulating natural wear and vertical masks mimic car occlusion and dashed lines.

datasets consist of limited lane situations compared to complex real-world scenes where the lanes are usually broken due to wear and occlusion. Therefore, we randomly mask the images using bar-like lines to mimic broken lanes. In Figure 2, the rectangle marked in red shows an example of a possible lane broken situation, a commonly seen dashed lane. Two gray bars are set in horizontal and vertical directions to simulate different types of occlusions. When masking with horizontal bars, the mask narrowly cuts the lane to represent natural wear. A larger area is blocked when setting vertically, this simulates car occlusions and dashed lines.

The masking is applied according to specific procedures. First, given an image I , the probability for it to be masked is p , and the rest of the images are skipped without masking with the probability of $1 - p$. Second, for the selected image, we randomly decide the orientation of the line-shaped mask, and the probability for the image I to be masked with horizontal lines and vertical lines are \hat{p} and $1 - \hat{p}$ respectively. Next, another random value n is generated to determine the number of masking lines. The created masking lines share the same width w , approximately the average lane line width in the image. With the generated lines, we erase the pixels in the line-shaped areas with the ImageNet [4] image mean values, a example of the line-shaped masks is shown in Figure 2.

4. Experiments

In this section, we evaluate the effectiveness of the proposed modules on three lane line detection benchmarks, TuSimple [1], LLAMAS [3], and CULane [17]. The following subsections present the datasets, evaluation metrics, implementation details, the performance comparison between our model and other published approaches, and the ablation study on the effects of the Large Kernel Enhanced Attention module and random Random Line Masking.

Algorithm 1: Pseudo code for random line erasing

```

Input: input parameters Image  $I$ 
Output: output result, Mask Image  $I$ 
Symbols: Masking probability  $p$ ; Probability to
horizontal lines  $\hat{p}$ ; Number of lines  $n$ 
 $n = \text{RandomNumber}$ 
if  $p < 0.5$  then
| return  $I$ ;
else
| if  $\hat{p} < 0.5$  then
| | Mask with  $n$  equidistant horizontal lines;
| else
| | Mask with  $n$  equidistant vertical lines;
| end
| return  $I$ ;
end
```

Dataset	Train	Val	Test	Resolution	Metrics
TuSimple [1]	3268	358	2782	720×1280	Acc. FPR. FNR.
CULane [17]	88880	9675	34680	590 × 1640	F1 Score
LLAMAS [2]	58269	20844	20929	717 × 1276	F1 Score Precision Recall

Table 1. The configuration for each datasets.

4.1. Datasets

To verify the generality of our proposed methods, we evaluate our approaches on TuSimple, LLAMAS [2], and CULane [17] datasets. The TuSimple contains highway images collected in good weather and traffic conditions with high quality. In addition, the LLAMAS [2] dataset, a new large-scale dataset, also consists of highway images. In the contract, the CULane [17] dataset has the largest scale road scene data and includes complex conditions such as night, crowd, and dazzling light. Table 1 demonstrates the details of these datasets.

4.2. Evaluation Metrics

The evaluation metrics for these three datasets are shown in Table 1.

The TuSimple evaluates model performance by accuracy (Acc), false positive rate (FPR), and false negative rate (FNR). The accuracy is computed as Eq. (3) shows. In TuSimple, we define the matched points as the predicted points whose Euclidean distance to the nearest ground truth points is less than 20 pixels.

$$Acc = \frac{N_m}{N_t} \quad (3)$$

N_m means the number of matched points, N_t means the

Method	CULane [17] [17]											TuSimple [1]			
	Total	Normal	Crowd	Night	No line	Shadow	Arrow	Dazzle light	Curve	Cross ↓	Acc.	FPR ↓	FNR ↓	params(M)	Flops(G)
Semantic Segmentation-based															
SCNN (ResNet-18) [17]	72.19	90.98	70.17	66.54	43.12	66.31	85.62	62.20	65.58	1808	94.77	0.075	0.074	12.42	89.34
SCNN (ResNet-34) [17]	72.70	91.06	70.41	67.75	44.64	68.98	86.50	61.57	65.75	2017	95.25	0.063	0.063	22.52	163.72
SCNN (ResNet-101) [17]	73.58	91.10	71.43	68.53	46.39	72.61	86.87	61.95	67.01	1720	95.69	0.052	0.050		
UFLD (ResNet-18) [21]	68.4	87.7	66.0	62.1	40.2	62.8	81.0	58.4	57.9	1743	—	—	—	—	—
UFLD (ResNet-34) [21]	72.3	90.7	70.2	66.7	44.4	69.3	85.7	59.5	69.5	2037	—	—	—	—	—
RESA (ResNet-18) [33]	72.90	91.23	70.57	67.16	45.24	68.01	86.56	64.32	66.19	1679	95.24	0.069	0.057	6.62	61.33
RESA (ResNet-34) [33]	73.66	91.31	71.80	67.54	46.57	72.74	86.94	64.46	67.31	1701	95.15	0.069	0.059	12.01	101.72
Instance segmentation based															
PINet(1H) [14]	69.4	85.8	67.1	67.1	44.8	63.1	79.6	59.4	63.3	1534	—	—	—	—	—
PINet(2H) [14]	73.8	89.6	71.9	67	49.3	67	84.2	65.2	66.2	1505	—	—	—	—	—
PINet(3H) [14]	74.3	90.2	72.4	67.7	49.6	68.4	83.6	66.4	65.4	1486	—	—	—	—	—
PINet(3H) [14]	74.4	90.3	72.3	67.7	49.8	68.4	83.7	66.3	65.6	1427	—	—	—	—	—
Anchor-based															
LaneATT (ResNet-18) [24]	74.88	90.98	72.78	68.61	48.23	69.68	85.44	65.43	63.18	1163	95.57	0.036	0.030	—	—
LaneATT (ResNet-34) [24]	76.42	91.94	74.76	70.32	49.17	77.68	88.14	65.92	68.07	1323	95.63	0.035	0.029	—	—
LaneATT (ResNet-122) [24]	76.79	91.50	76.04	70.43	50.29	75.96	86.16	68.99	63.99	1265	96.10	0.056	0.022	—	—
CurveLane-S [31]	71.4	88.3	68.6	66.2	47.9	68	82.5	63.2	66	2817	—	—	—	—	9
CurveLane-M [31]	73.5	90.2	70.5	68.2	48.8	69.3	85.7	65.9	67.5	2359	—	—	—	—	35.7
CurveLane-L [31]	74.8	90.7	72.3	68.9	49.4	70.1	85.8	67.7	68.4	1746	—	—	—	—	86.5
Curve-based															
LaneTrans [15]	—	—	—	—	—	—	—	—	—	—	93.69	0.098	0.072	—	—
PolyLaneNet (EfficientNet-B0) [25]	—	—	—	—	—	—	—	—	—	—	93.36	0.094	0.093	—	—
Least-Squares Fitting [25]	—	—	—	—	—	—	—	—	—	—	95.80	—	—	—	—
BézierLaneNet (ResNet-18)* [7]	72.37	89.20	70.86	66.47	43.79	67.97	82.93	64.28	55.61	1075	95.41	0.053	0.046	4.1	14.56
Ours	73.18	89.94	70.88	67.87	46.11	69.33	83.19	65.38	59.24	1200	95.95	0.046	0.027	4.82	15.94

Table 2. Results on *test* set of CULane [17] and TuSimple [1]. *: results reproduced by us, the best performance comes from 3 random runs.

Method	LLAMAS [2] (val) [2]				
	F1	Precision	Recall	params(M)	Flops(G)
Segmentation-based					
SCNN (ResNet-34) [17]	94.25	94.11	94.39	22.52	163.72
Anchor-based					
LaneATT (ResNet-18) [24]	93.46	96.92	90.24	12.02	—
LaneATT (ResNet-34) [24]	93.74	96.79	90.88	22.12	—
LaneATT (ResNet-122) [24]	93.54	96.82	90.47	—	—
Curve-based					
PolyLaneNet (EfficientNet-B0) [25]	88.40	88.87	87.93	—	—
BézierLaneNet (ResNet-18)	94.91	95.71	94.13	4.1	14.56
BézierLaneNet (ResNet-34)	95.17	95.89	94.46	9.49	29.85
Ours	96.21	96.52	95.90	4.82	15.94

Table 3. Comparison results on LLAMAS [2] validation set.

number of ground truth points.

The LLAMAS and CULane [17] use F1-score to evaluate the model performance, which is computed from the Precision and Recall:

$$Precision = \frac{N_m}{N_{pred}}, \quad Recall = \frac{N_m}{N_t} \quad (4)$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (5)$$

N_{pred} denotes the number of predicted points.

4.3. Implementation details

In order to fairly compare with other methods, we follow the previous training setting in [7] and implement our code based on PytorchAutoDrive [7] framework. The hyperparameters remain the same throughout all experiments. To be detailed, we set batch size equal to 20 and learning rate equal

to 0.00006 with Adam optimizer. Also, we adopt CosineAnnealing to decrease the learning rate during training. Due to the scale variation of different datasets, the training epochs in TuSimple, LLAMAS [2], and CULane [17] are 400, 36 and 20 respectively. Moreover, we adjust the resolutions for input images from different datasets into 360×640 for TuSimple and LLAMAS [2], and 288×800 for CULane. Regarding the detailed system setup, the probability p for an image I to be masked is set to 0.6, and the probability \hat{p} to apply horizontal lines is 0.5. The kernel size in the LKEA module is finally set to 11×11 .

4.4. Comparing with the state-of-the-art methods

Comparison on the TuSimple dataset. We first report our method performances on the TuSimple dataset in Table 2. Our method surpasses the baseline and outperforms BezierLaneNet with ResNet-34 backbone using 50% parameters, demonstrating that we are less suffering from overfitting on small-scaled TuSimple dataset. Also, it is worth noticing that our model achieves comparable results in contrast to a large model LaneATT [24] with ResNet-122 backbone. This indicates that the performance improvements are contributed from the efficient LKEA module of our model instead of only increasing model parameters.

Comparison on the CULane [17] dataset. On the challenging CULane [17] Dataset in Table 2, our method achieves the best total F1 scores among other curved-based methods. Moreover, our approach reaches the best performance and model size trade-off compared to all other methods. For example, compared with the segmentation-based method

LKEA	RLM	Acc.	FPR	FNR
		95.01	0.053	0.046
✓		95.80	0.052	0.032
	✓	95.57	0.056	0.037
✓	✓	95.95	0.046	0.027

Table 4. Modular ablation study on the TuSimple dataset. **LKEA**: Large Kernel Enhanced Attention module. **RLM**: Random lane masking.

Kernel size	Acc.	FPR	FNR
1 × 1	95.09	0.053	0.040
3 × 3	95.67	0.053	0.033
7 × 7	95.71	0.047	0.033
11 × 11	95.80	0.052	0.032

Table 5. Impact on performances with different kernel sizes in LKEA module.

SCNN (ResNet-34) [17], our approach obtains better results with only 20% parameter size. Achieving such results on this challenging dataset indicates the robustness of our system. **Comparison on the LLAMAS [2] dataset.** We also report our result on the latest large-scale dataset LLAMAS [2] as shown in Table 3, our method outperforms all other methods reported on the validation set. Table 3 indicates our method achieves better F1 scores with fewer parameters.

4.5. Ablation study

Modular ablation study. As demonstrated in Table 4, we validate the effectiveness of each module, the proposed Large Kernel Enhanced Attention module and Random Line Masking augmentation on LLAMAS [2] validation dataset. Table 4 shows that solely adding LKEA or Random lane Masking will improve the baseline performance, verifying the usefulness of each module. Only adding LKEA improves the accuracy from 95.01 to 95.80, which reflects the importance of global modeling ability. Combining LKEA and RLM, our model achieves the best accuracy 95.95.

Influence of the kernel size of the LKEA module. In Table 5, we investigate the impact of different kernel sizes in the LKEA module, including 1x1, 3x3, 7x7 and 11x11. As presented in Table 5, the model performance consistently improves as kernel size increases. And we hypothesize this result is led by two factors. The first one is larger kernel can have a longer shape bias which is also reflected in [5]. For lane lines with long and thin geometric shapes, owning such bias is necessary for a system to detect these lines. The second one is large kernel can bring more precision global position embedding, which is used by followed attention module. Precision localization is also important for the detect lines due to the pixel-level accuracy requirements in real applications. Here we clarify we don't test kernel sizes that are larger than 11x11, such as 31x31 kernel size used in [5] due to limited computation resources. However, we will conduct experiments on larger kernel sizes when we have more resources, and we believe current performance

still has room to be improved.

5. Limitations, Future Work and Discussions

We only experiment with our module and data augmentation on the latest BézierLaneNet architecture. However, according to the excellent performance of LKEA and RLM, we hypothesize that these two modules be a general performance-enhanced module that can also be plugged into other current lane detection frameworks. Furthermore, this module method may also be used in other types of methods, such as the segmentation based-method, instead of only the curve-based methods. We will conduct these experiments in the future.

Currently, we solely add two of our blocks in BézierLaneNet due to limited computation resources. As shown in the experiments part, two blocks bring noticeable results. Because of the global information grasping ability as well as the long-shape bias, we think this block can potentially be used as a basic block for designing a basic backbone used for car lane detection or similar tasks that require both long-range modeling and precision position awareness ability.

Although our proposed Random Lane Masking augmentation alleviates the overfitting problem by introducing more hard samples into the limited-size dataset, another common problem in lane detection that has still not been solved is the domain gap problem. In future work, we will propose an adaptive domain model by generating different weather-changing datasets or light-changing datasets using GANs, diffusion models, or image-relighting techniques. We hope the model can learn invariant features under datasets with variant conditions to alleviate the domain adaption problem. The basic idea is illustrated in Appendix A.2.

6. Conclusion

In this paper, we propose a novel Large Kernel Enhanced Attention module. The proposed LKEA module naturally owns global information modeling capability and position awareness by bridging the large kernel convolution and self-attention together. Besides, a random lane masking data augmentation is introduced. It can create many broken lanes through the random masking several horizontal or vertical bars area in a road scene image. Random lane masking augmentation alleviates the overfitting problems, which provides a specifically designed augmentation for lane detection tasks. By taking advantage of these two modules, our curved-based method achieves state-of-the-art performance among all existing curved-based methods and very competitive results in three datasets compared with all different approaches. It is also very lightweight, e.g., it achieves a better result than the segmentation-based method SCNN (ResNet-34) with only 21% parameters on CULane, TuSimple, and LLAMAS [2] datasets. This model can be deployed on edge-cut devices like some autonomous cars due to its good performance and model size trade-off.

References

- [1] Tusimple benchmark. <https://github.com/TuSimple/tusimple-benchmark>, 2017. 4, 5
- [2] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane markers using maps. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 4, 5, 6, 9
- [3] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane markers using maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 4
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 4
- [5] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11963–11975, 2022. 2, 3, 6
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2
- [7] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 5
- [8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 2
- [9] Noa Garnett, Rafi Cohen, Tomer Pe'er, Roei Lahav, and Dan Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2921–2930, 2019. 1, 2
- [10] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *proceedings of the european conference on computer vision (ECCV) Workshops*, pages 0–0, 2018. 1, 2
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3
- [12] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. 2
- [13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2
- [14] Yeongmin Ko, Younkwon Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 1, 2, 5
- [15] 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. 5
- [16] 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. 1, 2
- [17] 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. 1, 2, 4, 5, 6, 9
- [18] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 1, 2
- [19] 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. 2
- [20] Fabio Pizzati, Marco Allodi, Alejandro Barrera, and Fernando García. Lane detection and classification using cascaded cnns. In *International Conference on Computer Aided Systems Theory*, pages 95–103. Springer, 2019. 1, 2
- [21] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *European Conference on Computer Vision*, pages 276–291. Springer, 2020. 5
- [22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1
- [23] Ravi Kumar Satzoda and Mohan Manubhai Trivedi. Drive analysis using vehicle dynamics and vision-based lane semantics. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):9–18, 2014. 1, 2
- [24] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 294–302, 2021. 1, 2, 5, 9
- [25] 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. 1, 2, 5, 9
- [26] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

- [27] Qiangchang Wang, Tianyi Wu, He Zheng, and Guodong Guo. Hierarchical pyramid diverse attention networks for face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8326–8335, 2020. [2](#)
- [28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. [2](#)
- [29] Ze Wang, Weiqiang Ren, and Qiang Qiu. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv preprint arXiv:1807.01726*, 2018. [1, 2](#)
- [30] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2018. [2](#)
- [31] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In *European Conference on Computer Vision*, pages 689–704. Springer, 2020. [1, 2, 5](#)
- [32] Yuhui Yuan, Lang Huang, Jianyuan Guo, Chao Zhang, Xilin Chen, and Jingdong Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018. [2](#)
- [33] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3547–3554, 2021. [5](#)
- [34] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. [9](#)
- [35] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE transactions on vehicular technology*, 69(1):41–54, 2019. [1, 2](#)

Larger Kernel, Attention and data Matters: Make Curve-based Lane Detection Great again

Supplementary Material



A: Origin image

B: Winter style

C: Summer style

Figure 3. Examples of possible style transferred images.

A. Appendix

Method	LLAMAS [2]			
	Ep.	F1	Precision	Recall
Segmentation-based				
Baseline (ResNet-34)*	18	93.43	92.61	94.27
SCNN (ResNet-34) [17]*	18	94.25	94.11	94.39
Point detection-based				
LaneATT (ResNet-18) [24]**	15	93.46	96.92	90.24
LaneATT (ResNet-34) [24]**	15	93.74	96.79	90.88
LaneATT (ResNet-122) [24]**	15	93.54	96.82	90.47
Curve-based				
PolyLaneNet (EfficientNet-B0) [25]**	75	88.40	88.87	87.93
BézierLaneNet (ResNet-18)	20	94.91	95.71	94.13
BézierLaneNet (ResNet-34)	20	95.17	95.89	94.46
Ours	20	95.27	96.10	94.46

Table 6. Results from LLAMAS [2] test server.

A.1. Testing results on LLAMAS [2] dataset

We also report the testing result on the LLAMAS [2] dataset with the official test server, as listed in Table 6. Our result outperforms all previous curved-based methods on the testing set. These results further demonstrate the effectiveness of our proposed modules.

A.2. Future work: Image style transfer

Observing the current dataset, for each distinct location, they only collect one "location-condition" correspondence, such as "Road 1 - Sunny condition". However, in the real world, the autonomous driving system encounters multiple "location-condition" correspondences. For example, assume a car is moving on a road r under weather a condition w , we define the image captured by the front camera as I_r^w . However, this car may also move on the same road under another weather condition, denote as $I_r^{w'}$. The detection results of I_r^w and $I_r^{w'}$ may vary due to the domain gap problem, which makes the driving system unstable.

To tackle this problem, we adopt the style transfer technique [34] to generate new images with different environmen-

tal appearances at the same location. As shown in Figure 3, Figure 3 (b) in winter style and (c) in summer style are the images transferred from Figure 3 (a) given a template style image. We put the different styles of images into the training set. We believe that the model can have certain domain adaption abilities through training with the new multi-style dataset.