

# Lane Line Detection based on Parallel Spatial Separation Convolution

Xile Shen<sup>1</sup>

shenxl19@mails.tsinghua.edu.cn

Zongqing Lu<sup>1</sup>

luzq@sz.tsinghua.edu.cn

Youcheng Zhang<sup>1</sup>

youcheng17@mails.tsinghua.edu.cn

Jing-Hao Xue<sup>2</sup>

jinghao.xue@ucl.ac.uk

<sup>1</sup> Department of Electronic Engineering,  
Tsinghua University  
China

<sup>2</sup> Department of Statistical Science,  
University College London  
United Kingdom

---

## Abstract

One of the fundamental tasks in autonomous driving is lane line detection. We aim to improve detection accuracy in complex scenarios and strike a better balance between performance and complexity of lane line detection networks. To this end, we first propose Parallel Spatial Separation Convolution (PSS-conv), a new convolution operation built on a new parallel spatial convolution decomposition and a channel-weighted feature merging strategy, to aggregate the features obtained from decomposed convolution. Then, we propose Parallel Spatial Separation Convolution with Message-Passing (PSS-conv-MP), in which a new message passing module is added before feature merging to enable slice-by-slice information propagation. Based on the PSS-conv, PSS-conv-MP, residual connection and non-bottleneck design, we construct a new lane line detection network called Parallel Spatial Separation Network (PSSNet), which can handle challenging scenes like curve and obscured lane lines. Extensive experiments show that PSSNet can achieve a superior performance on the challenging lane line detection benchmark CULane.

## 1 Introduction

Lane line detection is one of the most fundamental and important tasks in autonomous driving. Only when lane markings are located and tracked accurately, can self-driving vehicles drive safely on the road. Due to strong feature extraction capabilities of convolution neural networks (CNNs), CNN-based lane line detection methods achieve high detection accuracy in various traffic scenes [6, 9, 12, 13, 14, 17, 22, 28, 32].

However, there are still many challenges in lane line detection. Firstly, it is still difficult to locate the lane lines when they are occluded by other objects, such as vehicles ahead, pedestrians beside and shadow of the buildings. Secondly, various challenging scenarios, such as curve lane lines, poor weather and low light condition also limit the capability of models [22]. Thirdly, CNN-based lane line detection methods need to be designed with an appropriate trade-off between detection accuracy and network complexity. Hence, we focus

on improving detection robustness and making a good trade-off between network performance and complexity.

In this paper, inspired by recent progress in lightweight network design and semantic segmentation [4, 18, 24], we design an effective lane line detection network with optimized structure and small amount of parameters aiming to tackle the challenges mentioned above. To reduce network complexity, we propose a parallel spatial separation convolution. To improve network’s ability of extracting lane line features, we use height-wise and width-wise convolutions respectively to extract underlying features, height-wise and width-wise message passing modules to enrich detailed features and a channel-weighted feature merging strategy to merge features.

The main novelty and contributions of this paper can be summarized as follows:

1) We explore a new decomposition of spatially separable convolution. A standard 2D convolution ( $k \times k$ ) is decomposed into height-wise ( $k \times 1$ ) convolution and width-wise convolution ( $1 \times k$ ) in parallel, which allows our network to better understand row and column information simultaneously. In this way, it can reduce width-wise information loss caused by the existing cascaded spatial separation convolution in which width-wise information is only extracted from the feature map produced by height-wise convolution. In addition, such a parallel decomposition can make parallel computing more convenient.

2) Based on the above-proposed parallel spatial convolution decomposition, we propose a new convolution operation called parallel spatial separation convolution (PSS-conv). It is also combined with a weighted feature merging strategy to make the lane line feature extraction more flexible and efficient, especially for complex scenes.

3) A new form of context message passing module is added to the basic PSS-conv to further enrich and update the width-wise and height-wise features (PSS-conv-MP). Then, we construct residual structures named parallel spatial separation non-bottleneck (PSS-nbt) and parallel spatial separation with message passing non-bottleneck (PSS-nbt-MP). Finally, we propose an effective lane line network (PSSNet) by stacking PSS-nbt and PSS-nbt-MP modules.

## 2 Related Work

### 2.1 CNN-based Lane Line Detection

CNN-based lane detection method was first proposed in [9]. Then, in [20], lane detection problem is regraded as an instance segmentation problem based on CNNs. In order to further improve the detection accuracy, SCNN [22] proposes a CNN-based information transferring module within feature maps in four directions by a cascading way. However, the amount of parameters and computation of SCNN is unbearable in real-world applications. Recently, RESA [32] presents a feature aggregator to enrich the lane line features after ordinary CNN feature extraction. However, RESA still needs a large number of parameters.

The methods above all require a complex feature extraction network as backbone. In order to reduce network complexity, Hou et al. [7] proposes a self-attention distillation (SAD) module to aggregate contextual information. IntRA-KD [8] uses knowledge distillation to attain lightweight and real-time models. However, the detection performance of these lightweight algorithms is unsatisfying in complex scenarios. Recently, CurveLaneNAS [60], a new method based on neural architecture search (NAS) was proposed to find a proper backbone and a feature fusion module, which achieves higher detection accuracy.

However, NAS is computationally expensive, requiring 5,000 GPU hours per dataset.

In summary, recent algorithms of lane line detection suffer from either high network complexity and computation or poor detection accuracy. Hence, in this paper, we propose an effective lane line detection network with an optimized structure, which has a better trade-off between detection accuracy and complexity.

## 2.2 Separable Convolution

Separable convolution is a popular way to reduce network parameters and achieve lightweight segmentation [10, 19]. There are two main types of separable convolutions: spatially separable convolution [19] and depth separable convolution [27]. Spatially separable convolution (asymmetric convolution) is a decomposition in the spatial dimension. Rengarajan et al. [25] employ spatially separable convolution kernels to correct motion distortions. Existing spatially separable convolution factorizes a standard 2D convolution ( $k \times k$ ) into a height-wise convolution ( $k \times 1$ ) followed by a width-wise convolution ( $1 \times k$ ). However, Jin et al. [10] shows that directly applying spatially separable convolution would cause significant information loss.

In this paper, we explore a new form of spatially separable convolution (PSS-conv). Different from the existing cascaded spatial separation convolution, PSS-conv is formed by a standard 2D convolution being decomposed into height-wise convolution and width-wise convolution in parallel and merging the features by channel-wise weights, as detailed in Section 3.1.

## 3 Proposed Method

### 3.1 Parallel Spatial Separation Convolution

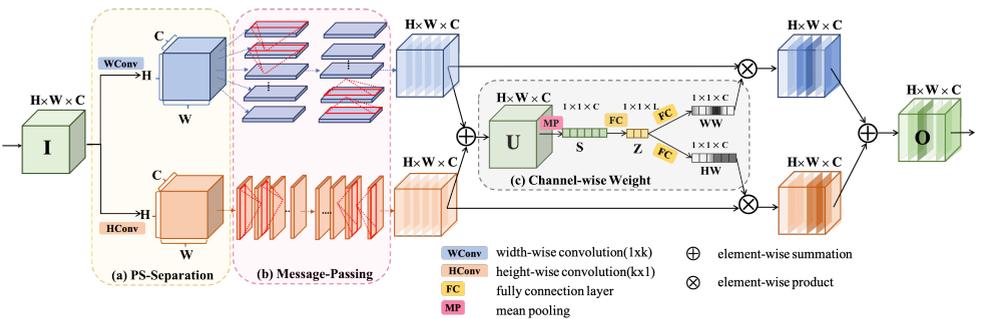


Figure 1: The diagram of PSS-conv and PSS-conv-MP. The Message-Passing module is only used in PSS-conv-MP.

To balance accuracy and efficiency in lane line detection, we design a new form of spatially separable convolution called PSS-conv, which is the fundamental module of our proposed network. Fig.1 excluding the Message-Passing module is the diagram of PSS-conv. Compared with the cascaded spatial separation convolution proposed in [10], our module can extract more low-level information by width-wise convolution and height-wise convolution

in parallel. Thus, our network is able to extract more spatial context information from original image at the same depth, which is crucial for a segmentation task such as lane line detection. In addition, PSS-conv has superior potential for parallelism to satisfy real-time detection.

As shown in Fig.1, PSS-conv can be divided into two simple steps (Fig.1(a), Fig.1(c)). Firstly, we use height-wise convolution ( $k \times 1$  kernel) and width-wise convolution ( $1 \times k$  kernel) separately to extract features (Fig.1(a)). Secondly, a channel-weighted feature merging strategy, motivated by SKNet [15], is used to merge the feature maps obtained by width-wise convolution and height-wise convolution (Fig.1(c)). This channel-weighted feature merging strategy uses the learned weights to achieve feature focusing based on different importance of width-wise and height-wise information at the channel level.

### 3.2 Parallel Spatial Separation Convolution with Message-Passing

In order to handle complex scenes, such as occlusion and dazzle light, we propose a spatial Message-Passing module (shown in Fig.1(b)) based on PSS-conv to gather more detailed information and enrich the feature map. As shown in Fig.1, the Message-Passing module can be divided to two branch following width-wise and height-wise convolutions respectively.

Specifically, assume we have a 3-D feature map tensor  $\mathbf{X}^w$  after width-wise convolution of size  $H \times W \times C$ , where  $H, W, C$  denote the number of rows, columns and channels respectively. It is noting that this tensor contains information within rows. Then, a height-wise Message-Passing module is used for feature enriching between rows. The tensor would be divided into  $H$  slices, and the first slice is sent into a convolution layer with  $C$  kernels of size  $1 \times \omega \times C$ , where  $\omega$  is the kernel width ( $\omega$  equals 7 in our experiments). The output is added to the next slice to provide a new one, and then this new slice is sent to the next convolution layer and this process would continue until the last slice is updated. Assume the 3-D kernel tensor  $\mathbf{K}^w$  with element  $K_{i,j,k}^w$  denoting the weight between an element in channel  $k$  of the last slice and current slice with an offset of  $j$  rows. Denote  $X_{i,j,k}^w$  as one of slices in 3-D tensor  $\mathbf{X}^w$  before the Message-Passing module, where  $i, j, k$  represent the index of row, column and channel. Then the forward computation of height-wise Message-Passing module is

$$X_{i,j,k}^{w'} = \begin{cases} X_{i,j,k}^w & j = 1 \\ X_{i,j,k}^w + f\left(\sum_m \sum_n X_{m,j-1,n}^{w'} \times K_{m,j,n}^w\right) & j = 2, 3, \dots, H \end{cases} \quad (1)$$

where  $f$  denotes the nonlinear activation function. The  $X$  denotes the input element, while  $X'$  denotes the element that has been updated. The convolution kernel weights are shared across all slices. In addition, height-wise Message-Passing module contains both downward and upward directions. Width-wise Message-Passing module is composed similarly but in the width direction.

Different from the message-passing module in SCNN [22] and RESA [23], we decouple the rows' and columns' message-passing. In this way, the combination of width-wise and height-wise information become more flexible, so the transmission of feature flow in our proposed method would be more benefit to lane line detection.

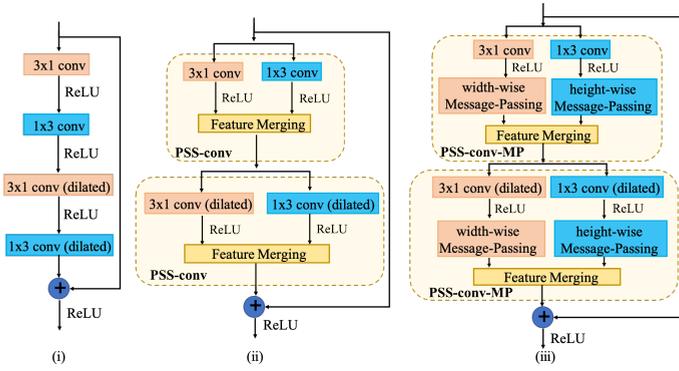


Figure 2: Three different residual layer modules: (i) the non-bottleneck-1D module in ERFNet; (ii) the PSS-nbt module; (iii) the PSS-nbt-MP module.

### 3.3 Lane Line Detection Network

#### 3.3.1 PSS-nbt and PSS-nbt-MP

Inspired by the non-bottleneck design in [5] and the non-bottleneck-1D module in ERFNet [26], we propose two new residual layers named PSS-nbt and PSS-nbt-MP based on PSS-conv and PSS-conv-MP respectively. As depicted in Fig. 2, they consist of two PSS-conv or PSS-conv-MP blocks: one is normal convolution while the other is dilated convolution to enlarge receptive fields. Moreover, residual connection between input and output of two PSS-conv or PSS-conv-MP blocks is used here to make full and efficient use of features.

#### 3.3.2 PSSNet

Fig. 3 shows the overall structure of our proposed PSSNet, where layers in Fig. 3(i) constitute the encoder producing downsampled feature maps, and layers in Fig. 3(ii) and Fig. 3(iii) constitute two decoding branches. The lane segmentation branch (Fig. 3(ii)) takes charge of the instance segmentation task, outputting probability maps of different lane line markings; and the lane line existence branch (Fig. 3(iii)) outputs a vector representing the existence of marked lane lines in the dataset. The lane line existence branch is used here to make convergence of network faster and more stable.

As shown in Fig. 3, PSSNet is achieved by sequentially stacking the proposed PSS-nbt and PSS-nbt-MP. In detail, the encoder in Fig. 3(i) includes 13 PSS-nbt units,  $N$  PSS-nbt-MP units and 3 downsampling units, which are performed by stacking parallel outputs of a single  $3 \times 3$  convolution with stride 2 and a max-pooling layer. The lane segmentation branch in Fig. 3(ii) includes 4 PSS-nbt units and 2 upsampling units. Different from the max-unpooling unit used for upsampling in SegNet [2] and ENet [23], the upsampling unit in our PSSNet is performed by a  $3 \times 3$  transposed convolution (deconvolution) with stride 2, which can reduce the memory usage as it needs no pooling index used in the encoder.

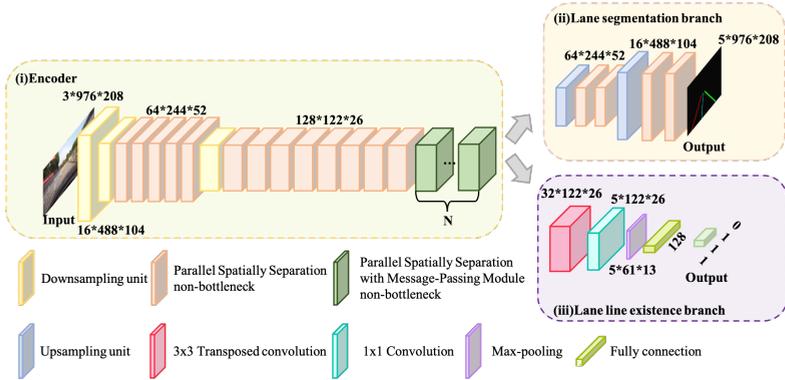


Figure 3: Overall architecture of the proposed PSSNet.

### 3.3.3 Loss Function

To train our complete network, we minimize the following loss function:

$$L = \lambda L_{seg} + \beta L_{exist} + L_{clDice}, \quad (2)$$

where  $L_{seg}$  is the weighted cross entropy loss commonly used in semantic segmentation tasks;  $L_{clDice}$  is the soft connectivity-preserving loss (clDice loss) which is first proposed in [11] for vessel segmentation;  $L_{exist}$  is the lane existence binary cross entropy loss for the lane line existence branch. Parameter  $\lambda$  is the weight for  $L_{seg}$  and  $\beta$  is the weight for  $L_{exist}$ . We empirically set  $\lambda = 0.4$ ,  $\beta = 0.1$ .

We introduce the clDice loss in lane line detection tasks for three reasons. Firstly, the positive and negative samples are extremely unbalanced in lane line detection, which would cause insufficient learning of positive samples and low detection accuracy. Secondly, the commonly used cross-entropy loss is calculated pixel by pixel, ignoring the inherent shape characteristics of the lane line markings. Thirdly, the clDice loss can directly optimize the network by an evaluation criterion which is similar to the F1 score.

In order to calculate the clDice loss, we first obtain the center line of the predict map ( $P$ ) and lane markings instance label map ( $L$ ) by iterative min- and max-pooling with a hyper-parameter  $k$ . The center line  $cl_P$  and  $cl_L$  are extracted from ( $P$ ) and ( $L$ ) respectively. Subsequently, we compute the fraction of  $cl_L$  that lies within  $P$ , which we call  $cl_L \circ lm_P$  and vice versa to get  $cl_P \circ lm_L$ .  $cl_P \circ lm_L$  is susceptible to false positives in the prediction while  $cl_L \circ lm_P$  is susceptible to false negatives. Therefore,  $cl_L \circ lm_P$  and  $cl_P \circ lm_L$  are regraded as precision and recall, respectively. Final expression of clDice is

$$clDice = 2 \times \frac{cl_L \circ lm_P \times cl_P \circ lm_L}{cl_L \circ lm_P + cl_P \circ lm_L}, \quad \text{where} \quad \begin{cases} cl_L \circ lm_P = \frac{|cl_L \cap P| + \varepsilon}{cl_L + \varepsilon} \\ cl_P \circ lm_L = \frac{|cl_P \cap L| + \varepsilon}{cl_P + \varepsilon} \end{cases}$$

and  $\varepsilon$  is used to keep the denominator from being 0. The final form of clDice loss used in our network is:

$$L_{clDice} = \sum_{c=1}^C 1 - clDice(P_c, L_c), \quad (3)$$

where  $C$  denotes the maximum number of lane lines in a single image, which is four in CULane. More experiments related to the cDice loss are in Section 4.4.3.

## 4 Experiments

In this section, we first introduce the dataset and the training setups used in the experiments. Then we compare our proposed method with other state-of-the-art methods. Finally, we investigate the effect of PSS-conv, Message-Passing module and cDice loss by ablation experiments.

### 4.1 Dataset and Training Settings

CULane is one of the largest and most complex datasets publicly available for multiple lane lines detection. In CULane, there are 88,880 images used for training, 9,675 for validation, and 34,680 for testing. All images are collected in Beijing’s urban roads with a resolution of 1,640x590 pixels; and all test images are divided into nine categories of challenging traffic scenarios (normal, crowded, night, cross, arrow, shadow, dazzle light, no visible line and curve). Before training, the upper part of images are cropped off because the information there is normally unrelated to the lane lines. After being cropped, these images are resized to 976x208. We pre-trained the segmentation branch on the Cityscapes dataset and initialized randomly for the lane existence branch. Our model is trained by SGD with batch size of 16 and an initial learning rate of 0.01 until convergence.

### 4.2 Evaluation Metrics

Following the official implementation of the evaluation on CULane [22], lane line markings are viewed as lines with widths equal to 30 pixels and then Intersection over Union (IoU) between the ground truth and the prediction is calculated. If a prediction has an IoU greater than 0.5 with a ground-truth lane, this predicted lane line would be considered as a true positive (TP). Then we employ  $F1 = \frac{2Precision \times Recall}{Precision + Recall}$  as the final evaluation metrics, where  $Precision = \frac{TP}{TP + FP}$ ,  $Recall = \frac{TP}{TP + FN}$  (FP is the false positive and FN is the false negative).

Method	Total	Normal	Crowded	Dazzle	Shadow	No Line	Arrow	Curve	Crossroad	Night	FLOPs (G)	Params (M)
SCNN [23]	71.60	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10	328.40	20.72
ENet-SAD [1]	70.80	90.10	68.80	60.20	65.90	41.60	84.00	65.70	1998	66.00	<b>3.90</b>	<b>0.98</b>
[24](Resnet-18)	68.40	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10	-	-
[25](Resnet-34)	72.30	90.70	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70	-	-
ERFNet [26]	73.10	91.40	71.50	65.60	71.50	45.20	87.10	66.00	2194	67.10	21.51	2.49
ERFNet-E2E [27]	74.00	91.00	73.10	64.50	74.10	46.60	85.80	<b>71.90</b>	2022	67.90	-	-
SIM-CycleGAN [28]	73.90	91.80	71.80	66.40	<b>76.20</b>	46.10	87.80	67.10	2346	69.40	-	-
CurveLanes-NAS-S [29]	71.40	88.30	71.80	63.20	68.00	47.90	82.50	66.00	2817	66.20	9.00	-
CurveLanes-NAS-M [29]	73.50	90.20	68.00	65.00	69.30	48.80	85.70	67.50	2359	68.20	35.70	-
CurveLanes-NAS-L [29]	74.80	90.70	70.50	67.70	70.10	49.40	85.80	68.40	1746	68.90	86.50	-
LaneATT (ResNet-18) [30]	75.09	91.11	72.96	65.72	70.91	48.35	85.49	63.37	<b>1170</b>	68.95	18.60	22.13
RESA (ResNet-50) [31]	75.30	92.10	73.10	<b>69.20</b>	72.80	47.70	88.30	70.30	1503	69.90	-	-
PSSNet (N=0)	74.60	92.10	72.50	65.70	70.80	47.00	87.50	69.30	1785	69.50	21.68	2.96
PSSNet (N=1)	<b>75.76</b>	<b>92.30</b>	<b>73.70</b>	68.10	72.80	<b>49.40</b>	<b>88.50</b>	70.80	1747	<b>71.20</b>	29.66	4.33

Table 1: Comparison of different algorithms on the CULane test set. F1 score is displayed except “Crossroad” for which only FP is shown. Top 1 results are in bold.

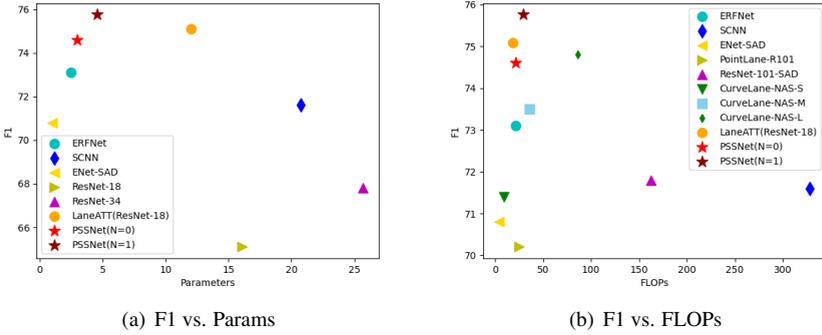


Figure 4: Trade-off comparison between the proposed PSSNet and other SOTA methods on the CULane test set.

### 4.3 Main Results

We compare our method with several state-of-the-art lane line detection methods. The results of these methods on nine categories of the CULane test set are shown in Table 1, in which PSSNet (N=0) and PSSNet (N=1) are the proposed network, with N being the number of PSS-conv-MP used between encoder and decoder. As Table 1 shows, our methods outperforms most models. For instance, PSSNet (N=0), which is only formed by PSS-conv, has comparable F1 score with CurveLane-NAS-L, while requiring only a quarter of FLOPs (Floating Point Operations). Moreover, PSSNet (N=1) obtains superior performance in most scenarios, strongly indicating the effectiveness and generality of our method.

Fig.4 presents a comprehensive performance comparison between ours and other methods. Fig.4(a) shows a trade-off comparison between performance (F1 score) and parameter amount; and the top-left corner indicates the best trade-off. It can be seen that our two methods (PSSNet (N=0) and PSSNet (N=1) represented by light red and dark red stars, respectively) are both located in the upper-left corner. In Fig.4(b), we compare more methods: PiontLane-R101 [24] and ResNet-101-SAD for completeness. Although LaneATT (ResNet-18) achieves higher F1 score and smaller amount of FLOPs than PSSNet (N=0), our method is superior to it in terms of parameter number.

Fig.5 compares the probability maps generated by ERFNet (the closest method to ours) and our method (PSSNet (N=1)). It can be seen that our predict probability maps are more accurate than those created by ERFNet, especially in some complex scenes such as dazzle light and night (rows 2, 3 and 4 of Fig.5). Fig.6 compares the detection results of ERFNet and our method (PSSNet (N=1)): the latter gets more blue (true-positive) lane lines and no red (false-positive) lane lines.

### 4.4 Ablation Study

#### 4.4.1 Effect of PSS-conv

In this section, we explore the effect of parallel spatial separation convolution. We compare the network formed by PSS-conv with classic semantic segmentation methods based on traditional convolution: ResNet-18 [25], ResNet-34 [26] and the ERFNet constituted by cas-

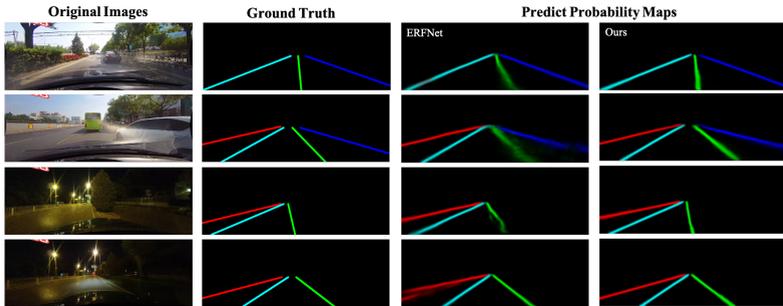


Figure 5: Probability maps of some images from the CULane test set. From left to right are the input images, ground truth, probability maps of ERFNet and our model.



Figure 6: Visual results of ERFNet (upper) and our method (lower) on the CULane test set. Green lines: ground-truth; blue line: true-positive; and red lines: false-positive.

caded spatial separation convolution. In this section, all networks are trained by the same cross-entropy loss. As Fig.2 shows, PSS-nbt has only half the depth of the non-bottleneck-1D in ERFNet. To complete comparison, we also construct a PSS-nbt that contains four PSS-conv blocks to keep the same depth as nonbottleneck-1D (Our\* shown in Table 2). As Table 2 shows, our method, with fewer parameters, can achieve better performance than classic semantic segmentation methods. Although ERFNet has fewer parameters than ours, our methods achieve better detection accuracy.

#### 4.4.2 Effect of Message-Passing module

In this section, we explore the effect of Message-Passing module. To compare fairly, we use the commonly used cross-entropy as loss function in this section. Results are shown in Table 3, where “w/o MP” represents our basic framework PSSNet (N=0) in this section, “w SCNN-MP” represents basic framework with “SCNN-DURL” [27] added between encoder and decoder, “w PSS-conv-MP once” and “w PSS-conv-MP twice” denote the basic framework with one and two PSS-conv-MP modules added before decoder respectively. Results show using our message passing module can make a significant improvement in detection accuracy. Compared with the message passing module proposed in [27], our PSS-conv-MP module can achieve better detection accuracy (Total Precision +1.2 and Total F1 score +0.4). In addition, experiments show that using PSS-conv-MP twice can only make a slight improvement in performance.

Method	Total F1	Params (M)
ResNet-18	65.10	16.06
ResNet-34	67.80	25.70
ERFNet	73.10	<b>2.49</b>
Ours	73.80	2.96
Ours*	<b>74.20</b>	5.29

Table 2: Performance of network formed by different convolutions. Cross-entropy loss is used in training.

Method	Total P	Total R	Total F1
w/o MP	74.47	73.22	73.84
w SCNN-MP	74.58	74.04	74.31
w PSS-conv-MP once	<b>75.77</b>	73.63	74.69
w PSS-conv-MP twice	75.11	<b>74.35</b>	<b>74.73</b>

Table 3: Performance of network with different message passing structures. Cross-entropy loss is used in training.

#### 4.4.3 Effect of cIDice Loss

In this section, we explore the effect of cIDice loss in lane line detection. We use the PSSNet (N=1) as basic framework in this section. Results are shown in Table 4. It can be seen that the detection performance can be significantly improved by using the cIDice loss. Additionally, we did some experiments on hyper-parameter  $k$  which determines the iteration numbers for getting the center line. The effect of different  $k$  applied to the label map can be visualized from Fig. 7. From left to right and top to bottom are the original label map and transformed map by  $k = 1, 2, 3$ . It shows that larger  $k$  produces thinner center line. Results in Table 4 show that  $k = 2$  is optimal for our task.

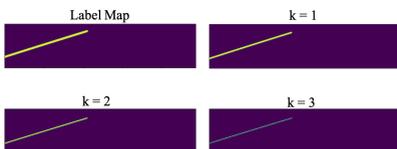


Figure 7: Effect of hyper parameters  $k$ .

Method	Total P	Total R	Total F1	
w/o cIDice loss	75.77	73.63	74.69	
w cIDice loss	k=0	75.78	73.78	74.77
	k=1	77.62	<b>74.84</b>	75.68
	k=2	<b>77.66</b>	73.95	<b>75.76</b>
	k=3	76.40	73.75	75.05

Table 4: Performance comparison of different  $k$  in cIDice loss.

## 5 Conclusion

In this paper, to maintain the detection accuracy as well as reducing model complexity, we first decompose a standard 2D convolution into height-wise convolution and width-wise convolution in parallel. Then, we use a channel-weighted feature aggregation to merge the features extracted by the decomposed convolution kernels. After that, to tackle the complex scenes, we propose PSS-conv-MP, in which a message passing module is added before feature merging in PSS-conv. Finally, we introduce PSS-conv and PSS-conv-MP to the residual layer and construct a new lane line detection network named PSSNet. On the CULane dataset, our network’s detection performance is comparable to the state-of-art methods but with less parameters, thus striking a superior balance between detection performance and efficiency for lane line detection. In addition, our module has superior potential for parallel computing, which is our future research direction.

## References

- [1] Jose Alvarez and Lars Petersson. DecomposeMe: Simplifying ConvNets for end-to-end learning. *arXiv preprint arXiv:1606.05426*, 2016.
- [2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE T-PAMI*, 39(12): 2481–2495, 2017.
- [3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE T-PAMI*, 40(4):834–848, 2017.
- [4] Zhenpeng Chen, Qianfei Liu, and Chenfan Lian. PointLaneNet: Efficient end-to-end cnns for accurate real-time lane detection. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2563–2568. IEEE, 2019.
- [5] 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.
- [6] Yuenan Hou. Agnostic lane detection. *arXiv preprint arXiv:1905.03704*, 2019.
- [7] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection CNNs by self attention distillation. In *ICCV*, pages 1013–1021, 2019.
- [8] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-region affinity distillation for road marking segmentation. In *CVPR*, pages 12486–12495, 2020.
- [9] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.
- [10] Weihao Jiang, Zhaozhi Xie, Yaoyi Li, Chang Liu, and Hongtao Lu. LRNNet: A lightweight network with efficient reduced non-local operation for real-time semantic segmentation. In *ICMEW*, pages 1–6, 2020.
- [11] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014.
- [12] Jihun Kim and Minho Lee. Robust lane detection based on convolutional neural network and random sample consensus. In *International conference on neural information processing*, pages 454–461. Springer, 2014.
- [13] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. VPGNet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 1947–1955, 2017.

- [14] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-CNN: End-to-end traffic line detection with line proposal unit. *IEEE T-ITS*, 21(1):248–258, 2019.
- [15] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, pages 510–519, 2019.
- [16] Tong Liu, Zhaowei Chen, Yi Yang, Zehao Wu, and Haowei Li. Lane detection in low-light conditions using an efficient data enhancement: Light conditions style transfer. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1394–1399. IEEE, 2020.
- [17] Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jih Lin. Multi-class lane semantic segmentation using efficient convolutional networks. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019.
- [18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.
- [19] Franck Mamalet and Christophe Garcia. Simplifying ConvNets for fast learning. In *ICANN*, pages 58–65, 2012.
- [20] 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.
- [21] Johannes C Paetzold, Suprosanna Shit, Ivan Ezhov, Giles Tetteh, Ali Ertürk, Helmholtz Zentrum Munich, and Bjoern Menze. cldice—a novel connectivity-preserving loss function for vessel segmentation. In *Medical Imaging Meets NeurIPS 2019 Workshop*, 2019.
- [22] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial CNN for traffic scene understanding. *arXiv preprint arXiv:1712.06080*, 2017.
- [23] 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.
- [24] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. *arXiv preprint arXiv:2004.11757*, 2020.
- [25] Vijay Rengarajan, Yogesh Balaji, and AN Rajagopalan. Unrolling the shutter: Cnn to correct motion distortions. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 2291–2299, 2017.
- [26] Eduardo Romera, José M Alvarez, Luis M Bergasa, and Roberto Arroyo. ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation. *IEEE T-ITS*, 19(1):263–272, 2017.
- [27] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for image classification. *Ph. D. thesis*, 2014.

- [28] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixão, Claudine Badue, Alberto F De Souza, and Thiago Olivera-Santos. Keep your eyes on the lane: Attention-guided lane detection. *arXiv preprint arXiv:2010.12035*, 2020.
- [29] Yu Wang, Quan Zhou, Jia Liu, Jian Xiong, Guangwei Gao, Xiaofu Wu, and Longin Jan Latecki. LEDNet: A lightweight encoder-decoder network for real-time semantic segmentation. In *ICIP*, pages 1860–1864, 2019.
- [30] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, and Zhenguo Li. CurveLane-NAS: Unifying lane-sensitive architecture search and adaptive point blending. *arXiv preprint arXiv:2007.12147*, 2020.
- [31] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-end lane marker detection via row-wise classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1006–1007, 2020.
- [32] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. RESA: Recurrent feature-shift aggregator for lane detection. *arXiv preprint arXiv:2008.13719*, 2020.