Multimedia Tools and Applications (2022) 81:21697–21717
https://doi.org/10.1007/s11042-022-12541-8

Direction-aware feedback network for robust lane detection

Jinhee Kim[1] · Wonjun Kim[1]

Received: 26 September 2021 / Revised: 20 December 2021 / Accepted: 31 January 2022 /
Published online: 16 March 2022

Abstract

Lane detection is a fundamental technique for autonomous driving systems. Various methods with deep neural networks have been actively introduced for this task, however, challenging issues, e.g., occlusion by vehicles, ambiguity by deterioration, etc., often give difficulties to accurately detect lanes in diverse road environments. To alleviate those problems, we propose the direction-aware feedback network. The key idea of the proposed method is to abundantly consider the global context of lanes by exploiting the directional attention module (DAM) in a multi-scale manner, which efficiently explores the high directionality with consideration of wide-range contextual dependencies both in horizontal and vertical directions. Moreover, such direction-aware features extracted from our DAMs are progressively refined by utilizing the feedback mechanism across different scale spaces, leading to the high-precision lane detection. Experimental results on benchmark datasets show the effectiveness of the proposed method under various road environments. The code and model are publicly available at: https://github.com/JinheeKIM94/Direction-aware_lane_detection

Keywords Lane detection · Direction-aware feedback network · Directional attention module · Wide-range contextual dependencies

1 Introduction

With the rapid growth of deep learning techniques in the field of computer vision, camera-based researches for autonomous driving systems have been actively introduced. Under such color-visible scenarios, lane detection has become a key technique since the corresponding result guides the direction of vehicles to go forward safely. This lane detection technique also can be a pre-requisite for further applications, e.g., traffic analysis by lanes, road defect

✉ Wonjun Kim
wonjkim@konkuk.ac.kr

Jinhee Kim
tyt8131@konkuk.ac.kr

[1] The Department of Electrical and Electronics Engineering, Konkuk University, Seoul, 05029, Republic of Korea

(such as crack and pothole) analysis by lanes, etc. To this end, there have been various attempts via the deep neural network (DNN) in recent days, however, most methods still struggle with lane detection due to harsh road conditions occurring in real-world environments, which make lanes invisible, e.g., occlusions by other vehicles or dimmed lanes by poor weather conditions.

Previous researches for lane detection have been conducted in two main directions, which are handcrafted feature-based approaches and learning-based approaches. In the former, low-level image features, e.g., color and edge properties of lanes [1, 8, 10, 41], were popularly employed with the Kalman filter [4, 16] and the Hough transform [5, 14]. To guide the process of lane detection more accurately, the additional supervisory information, e.g., vanishing point and semantic segmentation map, also has been combined with such low-level image features [7, 30, 37]. Even though handcrafted feature-based methods perform quite reliably under simple road environments, they often fail to grasp the semantic information to produce the consistent result in invisible or cluttered situations. In the category of learning-based approaches, DNN has been most widely adopted to determine whether each pixel belongs to the lane area or not. Specifically, DNN-based algorithms have often employed post-processing techniques, e.g., segment clustering and curve fitting, to refine the initial result of per-pixel classification [17, 28, 33]. Even though DNN-based methods significantly improve the performance of lane detection compared to handcrafted feature-based ones, the conventional feature extractor, which is generally composed of stacked convolutional layers, has its own limitation to consider the contextual information in a global sense. In order to involve the whole layout of the road environment in extracting lane areas, various architectures have been recently introduced with considerations of message passing, multi-task learning, etc. [15, 19]. On the other hand, several studies have started to concentrate on the real-time operation for autonomous driving under diverse real-world scenarios. For example, the concept of knowledge distillation is adopted to make the training model be light-weighted while reinforcing the representation power of the model from its own layers without external supervisors or labels [13]. Furthermore, the anchor point-based method significantly boosts up the inference speed by selecting the correct location of lanes at predefined row positions instead of segmenting every pixel of lanes [22]. However, those methods still have difficulties to detect thin or discontinuous lanes due to the limited range of the receptive field, which is highly related to lack of considering the wide-range dependencies in the activation results of the deep neural network.

In this paper, a novel yet simple method for robust lane detection is proposed. The key idea of the proposed method is to grasp the whole shape of lanes by exploiting directionally weighted features in a global manner. Since most lanes on real-world roads exhibit the continuous line structure, the direction-aware information can be usefully utilized to infer the whole shape of lanes even with insufficient appearance clues. Based on this observation, we design the directional attention module (DAM) to reinforce the learning process of the global context along principal directions (i.e., up, down, right, and left). Specifically, our DAM firstly explores the high directionality on the embedding space generated by the recurrent network, and then re-calibrates directionally weighted features by modifying the self-attentive weighting scheme [29] to consider a wide range of contextual dependencies. Moreover, we propose to apply the feedback mechanism to gradually refine the output of DAM across different scale spaces for precisely restoring the lane segmentation map. Experimental results on benchmark datasets show the efficiency and robustness of the proposed method for lane detection. The main contributions of this paper can be summarized as follows:

- We propose a novel way to exploit and highlight the directional property in the embedding space for understanding the global context of lanes even under complicated environments. To this end, the directional attention module (DAM) is devised and incorporated into the segmentation pipeline of the deep architecture while taking into account the wide-range contextual dependencies.
- We propose to refine directionally weighted features via the feedback mechanism across different scale spaces. Such refined features are gradually restored as the precise segmentation result, i.e., binary lane map.
- We evaluate the proposed method on three representative benchmark datasets, i.e., TuSimple [27], CULane [19], and BDD100K [36], and compare ours with state-of-the-art methods both qualitatively and quantitatively.

The remainder of this paper is organized as follows. A comparative review of related works is presented in Section 2. The proposed method is explained in detail in Section 3. Experimental results on benchmark datasets and ablation studies are reported in Section 4. The conclusions follow in Section 5.

2 Related work

In this Section, we present a comparative review of previous studies for lane detection, which can be divided into handcrafted feature-based methods and learning-based methods.

2.1 Handcrafted feature-based methods

Early works have attempted to model geometric characteristics of lane areas in a given scene by utilizing low-level image features. Specifically, Aly [1] first generated the top-view of the road image and then applied Gaussian filtering and thresholding to extract vertical lines as lanes. To estimate the dominant orientation of lanes, the voting scheme based on results of Gabor filtering was also introduced [41]. Choi et al. [8] considered the color characteristic of lanes, i.e., white and yellow attributes, and exploited the illumination-invariant color space to robustly detect such two colors even in complicated lighting conditions on the road environment. Satzoda and Trivedi [24] re-designed the top-view based filtering method to include the contextual information (e.g., road environments, traffic, etc.) for adaptively saving the computational power while keeping the accuracy of lane estimation. On the other hand, various line fitting techniques, e.g., Hough transform and spline, have been widely employed to refine predicted results of lane pixels as the post processing step. Borkar et al. [5] proposed to refine lane markers, which are detected in a similar way of [1], via their polar randomized Hough transform. Zhao et al. [40] used the Catmull-Rom spline, which is formed by arbitrary shapes of six control points, in combination with the extended Kalman filtering technique to accurately represent the curvature of lanes. More recently, many studies have concentrated on guiding results of lane detection by using the additional supervisory information. Among various supervisory candidates, the vanishing point has been most popularly employed as a good guider for lane markers due to the projective property of the road image, i.e., parallel lanes converge to the vanishing point in the projected image space. Ozgunalp et al. [18] proposed to extract the global optimum point from row-wise vanishing points and used it for multiple-curved lane detection on nonflat road surfaces. In [34], authors devised the probabilistic voting procedure with results of line segmentation to accurately extract the vanishing point in the background clutter and refined the result of lane detection by checking the location consistency of the vanishing point. Su et al. [26] proposed a stereo camera-based method for vanishing point constrained lane detection. On the other hand, Wu et al. [31] enhanced detection results of lane markings based on their distribution in the region close to the vehicle and further applied the partitioning technique with two simplified masks to such region for reduction of the computational burden. Even though such handcrafted feature-based methods have shown meaningful results for lane detection under limited road environments, those still fail to detect lanes with lack of visible clues due to occlusions by vehicles and dimmed lanes by deterioration.

2.2 Learning-based methods

With the development of the deep neural network (DNN) for scene understanding, many researchers have started to propose various learning-based methods to resolve the problem of lane detection. In the beginning, a simple encoder-decoder architecture has been adopted to generate the binary lane map from the color input, and the corresponding result was refined by post-processing techniques to accurately take into account for curved shapes of lanes. Specifically, Neven et al. [17] conducted instance segmentation based on two decoder branches, which were trained to cluster binary segmentation results by using the distance metric learning technique. After that, they train another neural network to predict parameters of the perspective transformation for the line fitting process. In [28], authors generated segmentation-like weight maps for each lane and subsequently used them to estimate parameters for curve fitting in a weighted least-square sense. On the other hand, several methods have been proposed to grasp the richer scene context with message passing or multi-task learning. Pan et al. [19] proposed to propagate the spatial correlation to the next layer via recurrent slice-by-slice convolutions for accurate lane segmentation. Lee et al. [15] introduced a unified architecture that jointly handles the problems of lane and road marking detections, which are guided by the vanishing point. In a similar way, Zhang et al. [38] jointly trained tasks for segmentations of lane boundaries and road areas to utilize the complementary relation of geometric constraints between these two tasks. More recently, light-weighted architectures have gained considerable attentions for lane detection in real-world scenarios. Hou et al. [13] designed the self-learning network with the concept of knowledge distillation, which was conducted by propagating the attention map extracted from the network itself to next low-level layers, to efficiently reduce the number of parameters. Qin et al. [22] only utilized the small number of anchor points at predefined row positions, thus the computational cost was significantly reduced compared to the case using the whole image for lane segmentation. Moreover, various domain concepts for lane detection also have been incorporated into learning-based methods. As an example, Yoo et al. [35] proposed to reformulate the problem of lane segmentation as the problem of row-wise classification. To do this, they compressed feature maps along the horizontal direction and represented each lane marker in rows. The technique of semantic segmentation can be directly applied to resolve the problem of lane detection. Specifically, the location-aware segmentation network allows for the spatial flow to integrate the object location into the segmentation pipeline, and shows the reliable result of road segmentation [6]. Gao et al. [9] proposed to learn the pixel-pair affinity in the pyramid network architecture, which gives the precise segmentation result for road components as well as general objects in a hierarchical manner. Furthermore, Zhang et al. [39] designed a sub-network to predict two-dimensional pixel embedding, which efficiently guides the output of instance segmentation by calculating a pixel's probability of the corresponding instance. This method provides reliable segmentation results for various objects in road scenes without the accurate location information for the bounding box. However, those methods still had a difficulty to detect whole lanes in occlusion situations by other vehicles because their networks are designed for per-pixel classification only on the visible parts. Different from previous segmentation-based approaches, we propose to exploit the global context of lanes in a direction-aware manner and refine such directionally weighted features with feedback operations across different scale spaces. Even though learning-based methods have accomplished the significant performance improvement for lane detection, they still suffer from thin and discontinuous lane structures due to the neglect of learning the global context.

In this paper, a novel yet simple method is proposed to consider the whole shape of lanes by exploiting directionally weighted features in a global manner. The feedback-based refinement process is also included in the proposed network. The technical details of the proposed method will be explained in the following Section.

3 Proposed method

The heart of the proposed method lies in learning the directionality of lanes to consider the global context of the road environment even under challenging conditions. To this end, we propose to design the directional attention module (DAM), which efficiently extracts features along with four principal directions in a recurrent architecture. It is noteworthy that such directionally weighted features are re-calibrated via the self-attentive process to allow for a wide range of contextual dependencies in our DAM. Finally, the output of DAM is progressively refined by feedback operations across different scale levels. The overall architecture of the proposed method is shown in Fig. 1.

3.1 Overview of the proposed architecture

The proposed network consists of three encoder-decoder streams to generate the lane segmentation map and one auxiliary branch to predict the existence of lane pixels as shown in Fig. 1. We adopt ERFNet [23] as our encoder, which has shown the reliable performance

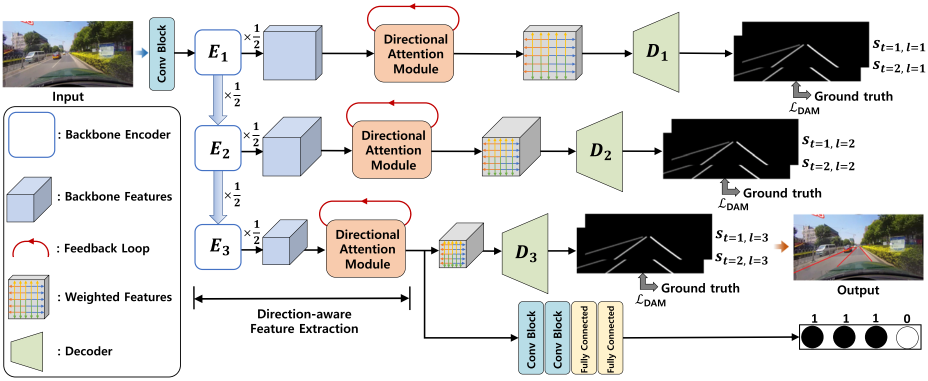

Fig. 1 Overall architecture of the proposed method for lane detection. The decoder for each scale space restores lane segmentation maps two times (i.e., $s_{t=1,2,\,l=1}$, $s_{t=1,2,\,l=2}$, and $s_{t=1,2,\,l=3}$ where t denotes the iteration index and l represents the scale index, respectively) due to the feedback operations conducted in DAMs. Note that $\mathcal{L}_{\text{DAM}}$ indicates the cross entropy-based loss function, which is defined in (2)

with the low computational cost for the segmentation task. Note that other backbone networks, e.g., VGG [25], ENet [20], ResNet [11], etc., also can be employed for the encoder of the proposed network. More concretely, features of different spatial scales, which are outputs of each encoder, i.e., E_1, E_2, and E_3 shown in Fig. 1, are fed into the proposed DAM. The spatial resolution of the feature map is reduced by a 1/2 scaling factor when passing through each encoder block. Such encoded features are subsequently re-calibrated according to the directional properties of lanes in a global sense. Note that details of the proposed DAM will be explained in the following subsection. Each decoder restores the lane segmentation map with the same resolution of the original input and the corresponding

Table 1 The detailed architecture of the proposed method

	Convolution block		
Convolution block type	Layer type	Weight dimension	Stride
Bottleneck	Conv	In×In×3 × 1	1
	Conv	In×In×1 × 3	1
	BatchNorm	-	-
	Conv	In×In×1 × 3	1
	Conv	In×In×3 × 1	1
	BatchNorm	-	-
Downsampler	Conv	In×Out×3 × 3	2
	BatchNorm	-	-
Upsampler	Deconv	In×Out ×3 × 3	2
	BatchNorm	-	-
Encoder (E_1, E_2, E_3) / Decoder (D_1, D_2, D_3)			
Module	Convolution block type	Channel	
Encoder	E_1	Downsampler	3/16
		Bottleneck	16/16
	E_2	Downsampler	16/64
		5×Bottleneck	64/64
	E_3	Downsampler	64/128
		8×Bottleneck	128/128
Decoder	D_3	Upsampler	128/64
		2×Bottleneck	64/64
		Upsampler	64/16
		2×Bottleneck	16/16
		Upsampler	16/N
	D_2	Upsampler	64/16
		2×Bottleneck	16/16
		Upsampler	16/N
	D_1	Upsampler	16/N

Note : the weight dimension is formulated as the number of input channels × the number of output channels × the height of the filter × the width of the filter. Channel indicates the number of input (In) and output (Out) channels for each convolution block. N denotes the total number of lanes. All convolution layers of the bottleneck block have the same input and output channels result is finally compared with the ground truth via the cross entropy-based loss function. As introduced in [19], the auxiliary network is incorporated into the proposed architecture for prediction of the lane existence, which is helpful for suppressing results falsely detected as lane pixels. The decoder is composed of several 3 × 3 deconvolution layers and 3 × 3 convolution blocks (i.e., convolution layer + batch normalization). Note that the number of layers in the decoder is determined according to the spatial resolution of direction-aware features, which are extracted from DAMs of different scale levels. The architecture details of the proposed method are also shown in Table 1.

Unlike previous methods, the proposed architecture contains feedback operations, that is, outputs for DAMs are returned back and concatenated with inputs of DAMs in three different scale levels as shown in Fig. 1. This feedback operation gradually refines the estimated result by clarifying lane areas as well as suppressing falsely detected pixels. Note that feedback operations are conducted two times and all the results are employed together for loss computation in our implementation.

3.2 Directional attention module

The key contribution of the proposed method is to consider directional properties in a global sense for lane detection as mentioned. To realize this process, we propose to design the directional attention module (DAM) as shown in Fig. 2. Motivated by the success of direction-aware shadow detection [12], we propose to adopt the two-round recurrent translation scheme along four principal directions, i.e., right, down, up, and left, for exploring the high directionality of lane features, which are extracted from the encoder, i.e., E_1, E_2, and E_3 shown in Fig. 1. Specifically, the contextual dependency both in horizontal and vertical directions are highlighted in the first round while such process is repeated based on the output of the first round to efficiently consider dependencies in the diagonal directions as well. As an example, we show one round of the recurrent translation to the right direction as follows [12]:

$$f(x, y) = \max\{\alpha_{right} f(x-1, y) + f(x, y), 0\}, \tag{1}$$

where $f(x, y)$ denotes the feature value at the (x, y) position. α_{right} denotes the weight value used in the recurrent translation layer for the right direction, which is initialized as

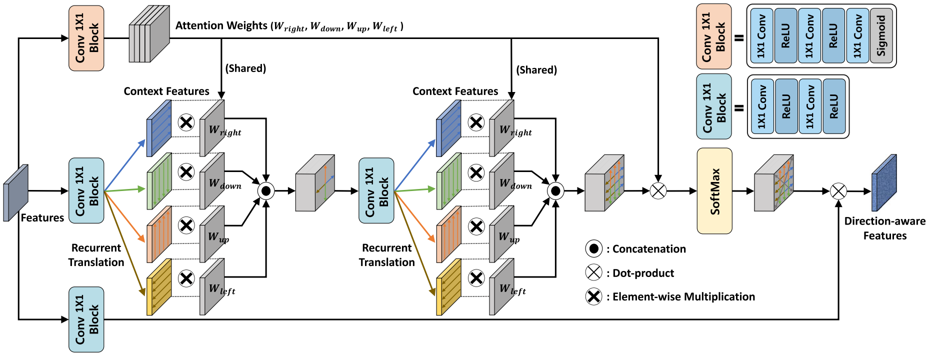

Fig. 2 The detailed architecture of the proposed directional attention module (DAM). Two rounds of recurrent translation are conducted with the direction-aware attention module along four principal directions. After this two-round process, such features are re-calibrated by the dot product operation with the attention weights, and the output of DAM is finally generated by the second dot product operation with input features. Note that attention weights are shared in two-rounds of recurrent translation

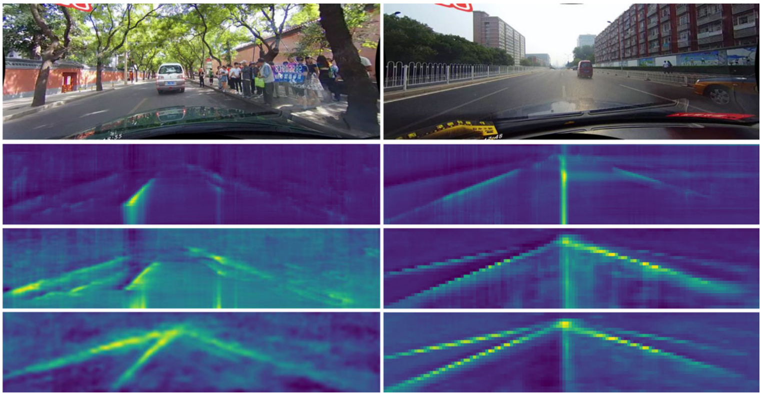

Fig. 3 From top to bottom: input images, outputs of the first round recurrent translation, outputs of the second round recurrent translation, and self-attentive results (i.e., final output of DAM). Note that bright values indicate high directionalities

the identity matrix and updated during the training phase (see [3] for more details). Note that this process is similarly conducted for other directions. In addition, attention weights, which are split into four channels, i.e., $\mathbf{W}_{right}$, $\mathbf{W}_{down}$, $\mathbf{W}_{up}$, and $\mathbf{W}_{left}$, are also simultaneously learned to enable the recurrent neural networks to selectively leverage spatial contexts aggregated along each direction for efficiently suppressing noisy responses as shown in Fig. 2. In order to allow for the wide-range dependency between distant lane pixels, we propose to slightly modify the self-attentive mechanism introduced in [29]. Specifically, we first compute the dot product of attention weights and directionally highlighted features instead of input features used in the original method. This dot product is fairly desirable to globally explore the whole shape of lanes with the guidance by the high directional responses. Finally, the output of DAM, i.e., direction-aware feature, is generated by the second dot product operation in a similar way of [29]. Note that attention weights are shared for the two-round recurrent translation process. Unlike [12] that has a limit to consider spatial connections between far distant pixels, we efficiently highlight the wide-range contextual dependencies along principal directions via the self-attentive mechanism, which leads to the reliable detection even with discontinuous lanes. Some visualization examples for directionally weighted features in the proposed DAM are shown in Fig. 3. We can see that high directionalities are well revealed on whole areas of lanes by using our DAM.

One important advantage of the proposed method is that the output of DAM is progressively refined by feedback operations. By involving the high-level semantics with input features, the learning process for DAM is guided towards revealing directional structures more accurately even with ambiguities driven by occlusions and background clutters. Note that the feedback operation is conducted two times in our implementation and the performance variation according to the number of feedback operations will be analyzed in the ablation study of Section IV. The advantage of the feedback operation is shown in Fig. 4. As can be seen, the lane pixels are successfully restored while false positives are significantly suppressed step by step through feedback operations.

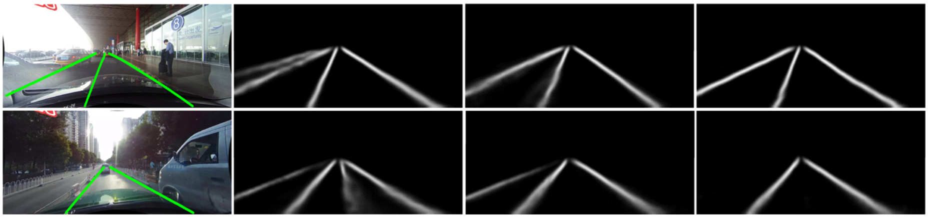

Fig. 4 Examples of the refinement process through feedback operations. 1st column: input images (green lines represent the ground truth). 2nd column: lane detection results without the feedback operation. 3rd and 4th columns: lane detection results with one feedback and two feedbacks, respectively

3.3 Loss functions

The proposed network is trained based on two types of loss functions. First, $\mathcal{L}_{\text{DAM}}$ determines whether each pixel belongs to the lane area or not, which is defined based on the cross entropy as follows:

$$\mathcal{L}_{\text{DAM}} = -\sum_{l=1}^{L}\sum_{t=1}^{T}\sum_{x,y} y(x,y)\log(s_{t,l}(x,y)), \tag{2}$$

where t and l denote the indices of feedback operations and scale spaces, respectively (thus, T and L are the total number of feedback operations and scale spaces, respectively). For the default setting, $T = 2$ and $L = 3$ are used. In the t-th feedback operation, $s_{t,l}(x, y)$ indicates the estimation result of the lane probability at the pixel position (x, y) of the l-th scale space. It should be emphasized that estimation results obtained from all the scales and feedback operations are used together for loss computation. $y(x, y)$ denotes the value of the ground truth, i.e., $y(x, y) = 1$ (lane pixel) or 0 (background), at the pixel position (x, y).

To reduce the number of pixels falsely detected as lane pixels, the binary cross entropy to check the index of the lane is adopted as the second loss function as follows [13]:

$$\mathcal{L}_{\text{EXT}} = \sum_{i=1}^{N} -p_i \log(g_i) - (1 - p_i)\log(1 - g_i), \tag{3}$$

where p_i and g_i denote the existence of the i-th lane and the corresponding probability estimated from the auxiliary network as shown in Fig. 1, respectively. N is the total number of lanes, which is automatically determined according to the training image. By combination of such two loss functions, the trainable parameters of the proposed network are efficiently optimized for lane detection as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{\text{DAM}} + \lambda\mathcal{L}_{\text{EXT}}, \tag{4}$$

where λ is the balancing factor and is set to 0.1 in our implementation.

4 Experimental results

In this Section, we demonstrate the robustness and efficiency of the proposed method through various experimental results based on three representative benchmark datasets, i.e., TuSimple [27], CULane [19], and BDD100K [36] datasets. Specifically, we firstly explain the implementation details of the proposed method. The performance of the proposed method is analyzed based on such benchmark datasets in detail and compared with state-of-the-art methods both qualitatively and quantitatively. The ablation study is finally conducted to show the role of each component in the proposed network.

4.1 Implementation details

The proposed method is implemented on the PyTorch framework [21] with the Intel i7-68850K@3.60GHz CPU and two NVIDIA GTX Titan XP GPUs. The stochastic gradient descent (SGD) is used to tune all the parameters of the proposed network with the batch size of 8 where the momentum and weight decaying factor are set to 9×10^{-1} and 1×10^{-4}, respectively. Note that the proposed network is trained for 96 epochs on the TuSimple dataset, and 32 epochs on the CULane and BDD100K datasets. The initial learning rate is set to 0.001 and adaptively changed to $0.001 \times (1 - \text{current epochs/total epochs})^{0.9}$ during training. We resize all the samples of TuSimple, CULane, and BDD100K datasets to 368×640, 288×800, and 360×640 pixels, respectively. Then, the technique of data augmentation, e.g., random crop and rotation, is subsequently applied to resized images for alleviating the overfitting problem. In regard to the processing time, it takes about 20 hours for training the proposed network with 88,880 samples of the CULane dataset, and the inference time is averagely about 50ms.

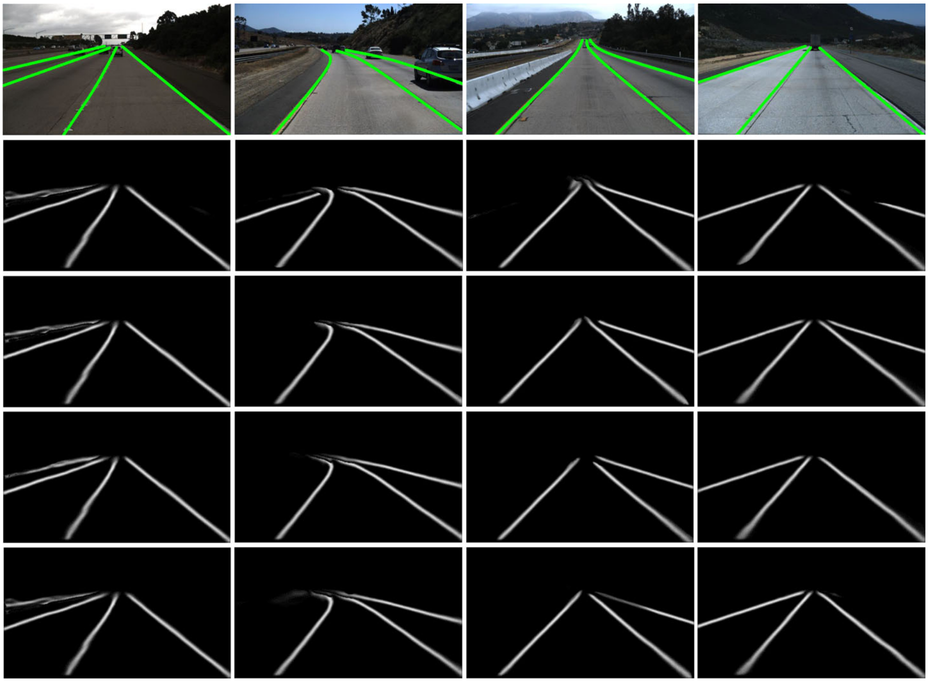

Fig. 5 Some results of lane detection in the TuSimple dataset. From top to bottom : input images with the ground truth (green color), results by ENet [20], SCNN [19], SAD [13], and the proposed method

4.2 Benchmark datasets

First of all, the TuSimple dataset [27] is composed of 6,408 frames, which are acquired from the highway scene. Even though this dataset contains relatively easy road environments, it is a pioneering dataset for learning-based lane detection. The TuSimple dataset is divided into three groups, i.e., 3,268 frames for training, 358 frames for validation, and 2,782 frames for test. On the other hand, the CULane dataset [19] consists of 133,235 frames, which are extracted from 55 hours of driving videos taken in urban, rural, and highway environments. Specifically, 88,880 and 9,675 frames are used for training and validation, respectively, while the test is performed with 34,680 frames. In particular, samples obtained from nine different challenging scenarios, i.e., normal, crowd, highlight, shadow, arrow, curve, cross, no line, and night, are tested for evaluating the performance in-depth. Lastly, the BDD100K dataset [36] provides 80,000 frames of driving scenes, and then such frames are newly divided into training (60,000 frames), validation (10,000 frames), and test (10,000 frames) subsets in a similar way of [13]. Note that the auxiliary branch of the proposed network to differentiate lane instances is not used for the BDD100K dataset since multiple lanes with more than eight are usually very close to each other.

4.3 Performance evaluations

Qualitative evaluation: To show the efficiency and robustness of the proposed method, we compare ours with several representative methods for lane detection, which are SCNN [19], SAD [13], and E2E-LMD [35]. In addition to this, two backbone networks most widely

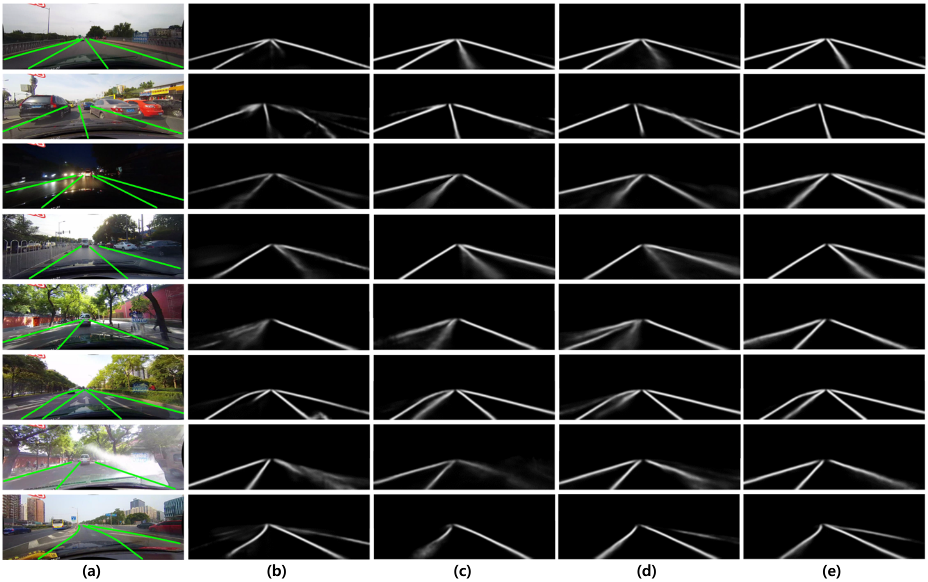

Fig. 6 Some results of lane detection according to eight different scenarios of the CULane dataset. Note that results for the "Cross" scenario are excluded in this example since it does not contain lanes (see Table 3 for checking the accuracy). From top to bottom (scenarios) : Normal, Crowd, Night, No line, Shadow, Arrow, Highlight, and Curve. **a** Input images with the ground truth (green color). **b** Results by ENet [20]. **c** Results by SCNN [19]. **d** Results by SAD [13]. (e) Results by the proposed method

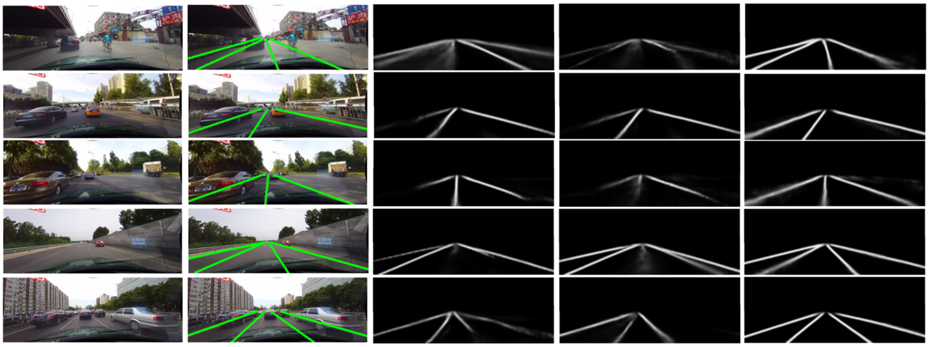

Fig. 7 Some results of lane detection for challenging scenarios in the CULane dataset. 1st column: input images. 2nd column: input images (green lines represent the ground truth). 3rd column: results by SCNN [19]. 4th column: results by SAD [13], 5th column: results by the proposed method

employed for image recognition and segmentation, i.e., ResNet [11] and ENet [20], are also compared with the proposed method as baseline models. Note that the simple decoder, which is composed of three deconvolution layers, is combined with these two backbone networks for generating the lane segmentation result. First of all, some results of lane detection in the TuSimple dataset are shown in Fig. 5. Specifically, most methods provide reliable results in the TuSimple dataset since samples are acquired in the highway scene with the relatively simple background. In Fig. 6, lane detection results in the CULane dataset are demonstrated according to different scenarios. Since "Cross" scenario does not contain lanes, we just report the accuracy in Table 3. As shown in Fig. 6, previous approaches often fail to accurately detect the overall shape of lanes due to occlusions and clarity deteriorations at the outer boundary. In contrast, the proposed method successfully extracts lanes by considering directionalities in a whole lane areas with feedback operations. Owing to such an ability to grasp the global context, the proposed method shows the good performance even

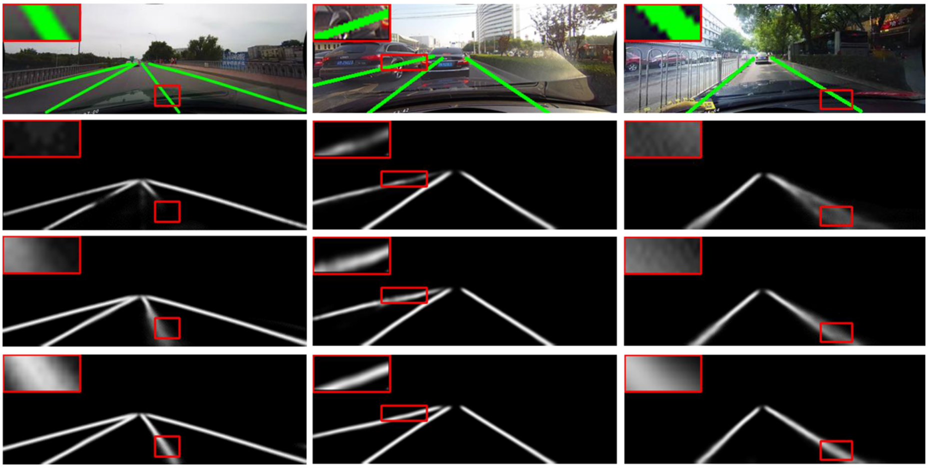

Fig. 8 Some results of lane detection in the CULane dataset. From top to bottom : input images with ground truth (green color), results by SCNN [19], SAD [13], and the proposed method. Note that the part of the lane region (i.e., red rectangle) is enlarged in the left corner

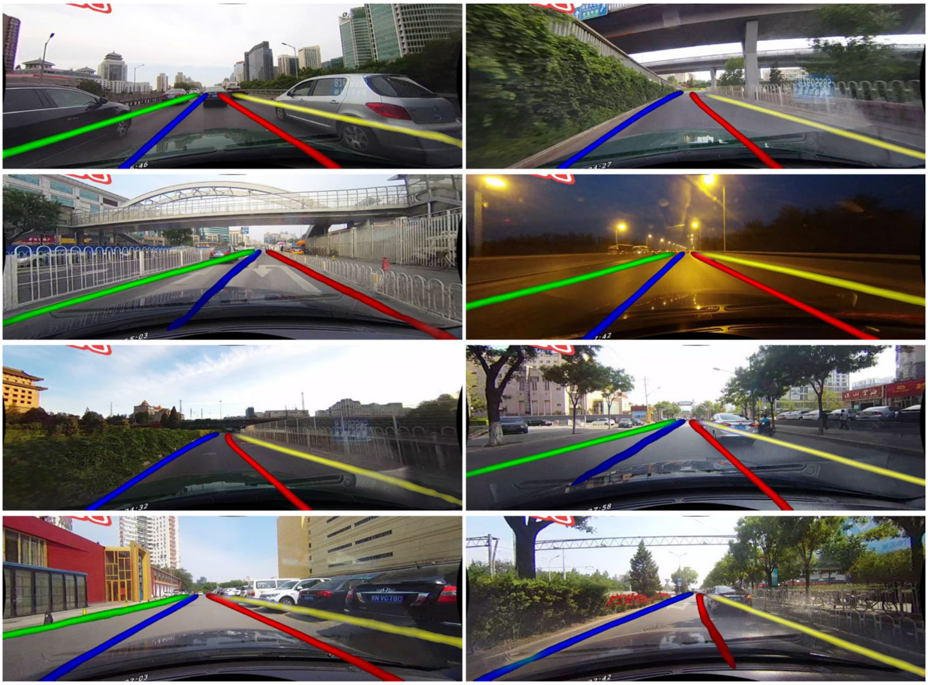

Fig. 9 Some examples of instance classification for lanes detected in the CULane dataset. Note that differently recognized lanes are represented with different colors

with heavily crowded scene as well as dim lanes (see the second, third, and fourth examples of Fig. 6e). In addition, we provide additional lane detection results for challenging scenarios in Fig. 7. Specifically, the proposed method shows accurate results for lane detection even in occlusions by vehicles or road conditions with dimmed lanes. Furthermore, some experimental results with enlarged local regions are presented in Fig. 8. As shown in Fig. 8, the proposed method successfully detects lanes in various road environments compared to previous methods. Specifically, our method shows accurate detection results even in occlusion by vehicles (see the second column of Fig. 8). Moreover, several classification results for detected lanes are also presented with different colors in Fig. 9. Since the proposed network contains the auxiliary branch for instance classification of lanes, it is helpful for the traffic and road analysis according to the lane. Finally, we provide lane detection results for the BDD100K dataset in Fig. 10. Test samples from this dataset contain more challenging conditions, e.g., split lanes, complicated illuminations, etc., thus the overall performance of lane detection is not good enough compared to previous two benchmark datasets. Nevertheless, the proposed method shows the reliable results compared to previous methods as shown in the last row of Fig. 10.

Quantitative evaluation : For the quantitative evaluation, we basically utilize the same metrics introduced in previous approaches. By following [13], we also tested the proposed method with different backbones, i.e., ResNet-18, ResNet-34, and ENet. Specifically, the detection accuracy shown in Table 2 denotes the hit rate, i.e., the ratio of truly detected pixels and the ground truth. FP and FN denote the false positive and the false negative, respectively.

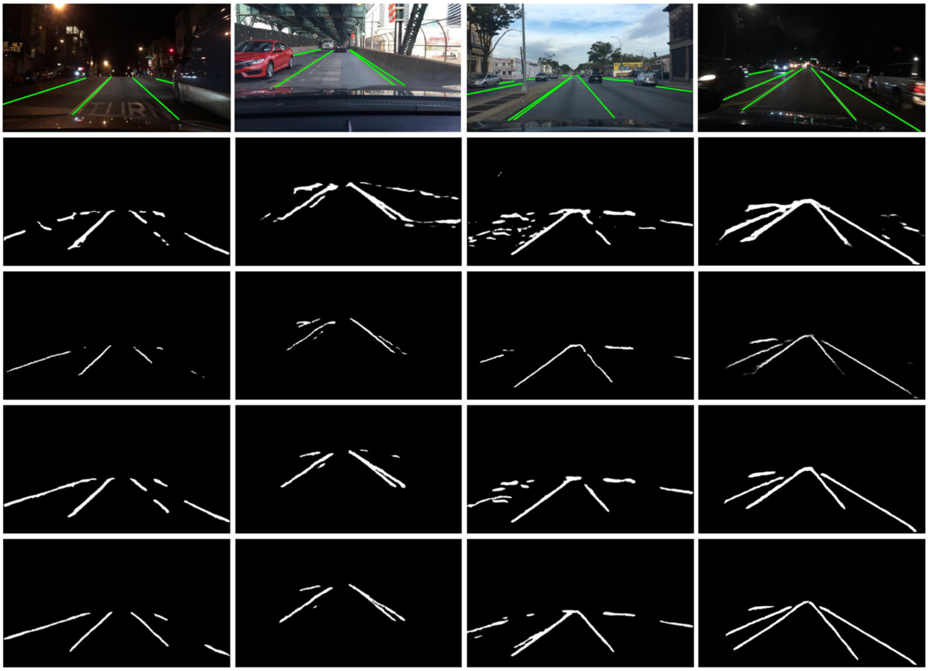

Fig. 10 Some results of lane detection in the BDD100K dataset. From top to bottom : input images with the ground truth (green color), results by ENet [20], SCNN [19], SAD [13], and the proposed method

As can be seen, the proposed method achieves the competitive performance in the TuSimple dataset. Since test samples contained in the TuSimple dataset are relatively simple (see Fig. 5), the performance of lane detection is almost saturated. The performance comparison on the CULane dataset is also shown in Table 3. As introduced in [19], we consider each lane marking as a line whose width is 30 pixel and compute the intersection over union (IoU) between estimated results and the ground truth. Note that estimated results whose IoU values are larger than 0.5 are used as true positives (TP). Based on this, F_1 score is

Table 2 Performance comparisons of lane detection based on the TuSimple dataset

Algorithm	Accuracy	FP	FN
ResNet-18 [11]	92.69%	0.0948	0.0822
ResNet-34 [11]	92.84%	0.0918	0.0796
ENet [20]	93.02%	0.0886	0.0734
SCNN [19]	96.53%	0.0617	0.0180
ENet-SAD [13]	96.64%	0.0602	0.0205
E2E-LMD [35]	96.02%	0.0321	0.0428
ResNet-18-Ours	95.87%	0.0524	0.0368
ResNet-34-Ours	96.15%	0.0466	0.0318
ENet-Ours	96.10%	0.0480	0.0322
ERFNet-Ours	96.23%	0.0437	0.0285

Table 3 Performance comparisons of lane detection based on the CULane dataset

Category	Proportion	SCNN [19]	ENet-SAD [13]	E2E-LMD [35]	R-18-Ours	R-34-Ours	ENet-Ours	ERFNet-Ours
Normal	27.7%	90.6	90.1	91.0	90.2	90.5	90.4	91.2
Crowd	23.4%	69.7	68.8	73.1	69.5	70.2	69.8	73.5
Night	20.3%	66.1	66.0	67.9	65.2	65.4	66.2	67.5
No line	11.7%	43.4	41.6	46.6	43.3	43.6	43.5	46.8
Shadow	2.7%	66.9	65.9	74.1	63.8	68.3	66.7	74.2
Arrow	2.6%	84.1	84.0	85.8	83.5	83.8	84.3	85.0
Highlight	1.4%	58.5	60.2	64.5	57.3	57.7	57.6	63.9
Curve	1.2%	64.4	65.7	71.9	67.9	67.5	68.8	71.5
Cross	9.0%	1990	1998	2022	2266	2290	2319	2004
Total	–	71.6	70.8	74.0	71.0	71.9	71.7	74.3

Note : F_1 scores are reported except “Cross” for which only FP is shown. The right part of this Table shows the performance variation of the proposed method according to different backbone networks, e.g., ResNet-18, ResNet-34, ENet, and ERFNet

Table 4 Performance comparisons of lane detection based on the BDD100K dataset

Algorithm	Accuracy	IoU
ResNet-18 [11]	30.66%	11.07
ResNet-34 [11]	30.92%	12.24
ResNet-101 [11]	34.45%	15.02
ENet [20]	34.12%	14.64
SCNN [19]	35.79%	15.84
ENet-SAD [13]	36.56%	16.02
ResNet-18-Ours	32.23%	13.81
ResNet-34-Ours	33.57%	14.63
ENet-Ours	36.98%	16.26
ERFNet-Ours	37.91%	17.04

computed as $\frac{2\times Precision\times Recall}{Precision+Recall}$ where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$. From Table 3, we can see that the proposed method provides the reliable results under various scenarios compared to previous methods. In particular, it is thought that our directional attention module (DAM) is effective for occlusions and loss of lanes (see “Crowd” and “No line” scenarios in Table 3) due to the ability to consider the global context based on directionalities. Finally, the performance on the BDD100K dataset, which is most recently published dataset, is demonstrated in Table 4. In contrast to previous datasets, it contains more challenging conditions, e.g., diverse weather conditions, double white lanes, etc., thus the overall performance of lane detection methods is significantly dropped. Among them, the proposed method shows the best performance in terms of the pixel-wise IoU metric. Based on various evaluation results, it is naturally thought that the proposed method works reliably under diverse road environments.

4.4 Ablation study

In this subsection, we investigate the performance variation according to different settings of the proposed architecture. Specifically, we first verified the advantage of the proposed DAM for lane detection as shown in Table 5. In Table 5, the baseline model represents the network, which utilizes only the backbone network (i.e., ERFNet) in the encoder part without DAM and feedback operations. Also, the ‘+ Feedback’ model conducts two times feedback iterations via simple convolution blocks (i.e., one 1×1 convolution and three 3×3 convolution

Table 5 Performance analysis of the proposed method according to changes of network architectures

Architectures	TuSimple	CULane	BDD100K	
	Acc.	F_1-score	Acc.	IoU
Baseline	95.18%	73.1	35.38%	15.65
+ DAM	96.03%	74.0	37.37%	16.71
+ Feedback	95.46%	73.5	36.12%	15.98
+ DAM + Feedback	96.23%	74.3	37.91%	17.04

Note : baseline indicates the architecture without DAM and feedback loops in Fig. 1

Table 6 Performance analysis of the proposed method according to the change of the directional attention module

Architectures	TuSimple	CULane	BDD100K	
	Acc.	F_1-score	Acc.	IoU
Baseline	96.09%	74.1	37.64%	16.82
+ Self-attentive weighting	96.23%	74.3	37.91%	17.04

Note : baseline indicates the directional attention module without self-attentive weighting in Fig. 2

layers), which are replaced with DAM, in different scale spaces. From Table 5, we can see that our DAM plays an important role to accurately detect lane pixels. The feedback loop, which is defined for DAM, is also helpful to improve the performance of lane detection. In the following, the effect of self-attentive weighting in DAM is tested and the corresponding result is shown in Table 6. In Table 6, the baseline model indicates the network where two dot product operations shown in Fig. 2 are removed from DAM. Based on the meaningful improvement, it is thought that directionally highlighted responses are effectively refined to restore the whole shape of lane areas based on the proposed dot operation. Additionally, we provide more detailed quantitative results in challenging scenarios where occlusions or loss of lanes occur frequently as shown in Table 7. As can be seen in Table 7, the proposed method shows the better performance than the reference spatial module ("Baseline" in Table 6), thus it is thought that our DAM is effective for lane detection.

Moreover, we explore the optimal number of feedback operations for our DAM as shown in Table 8. Interestingly, the large number of feedback operations slightly drops the performance. This is because parameters in DAM are probably overfitted as the feedback process is repeated too much. From Table 8, we conduct feedback operation two times, i.e., $T = 2$ as shown in (2). In addition, we tested the proposed method using a more effective loss function, which assigns a higher weight to the loss value when the feedback iteration increases, and the corresponding results are shown in Table 9. Since refined features have higher weights, the performance is improved while efficiently overcoming the limitation caused by

Table 7 Performance analysis of the proposed method according to changes of network architectures in challenging scenarios of the CULane dataset

Architectures	Category	
	Crowd	No line
	F_1-score	
Baseline	71.6	45.1
+ DAM (w/o self-attentive weighting)	72.9	45.9
+ DAM (with self-attentive weighting)	73.4	46.6
+ Feedback	72.2	45.4
Full model (ours)	73.5	46.8
ENet-SAD [13]	68.8	41.6
E2E-LMD [35]	73.1	46.6

Note : baseline indicates the architecture without DAM and feedback loops in Fig. 1

Table 8 Performance analysis of the proposed method according to the number of feedback operations

# of feedbacks	TuSimple	CULane	BDD100K	
	Acc.	F_1-score	Acc.	IoU
$T = 1$	96.03%	74.0	37.37%	16.71
$T = 2$	96.23%	74.3	37.91%	17.04
$T = 3$	96.21%	74.2	37.84%	16.91
$T = 4$	96.22%	74.1	37.90%	17.01
$T = 5$	96.19%	74.2	37.85%	16.95

the unbalanced distribution of lane pixels. Furthermore, the performance variation according to the different number of the scale spaces in the encoder is evaluated and the corresponding result is also shown in Table 10. It can be seen that the performance of lane detection gradually improves as the number of scale spaces increases. Since the performance improvement is not noticeable compared to the increase in the number of parameters between three and four levels, we use the backbone encoder of three levels as shown in Fig. 1, i.e., E_1, E_2, and E_3, which can still achieve the best performance for CULane and BDD100K datasets compared to previous methods. Additionally, we conducted additional experiments to check the effect of recurrent translations along each direction (i.e., right, down, up, and left) in the directional attention module as shown in Table 11. From Table 11, we can see that the recurrent translation process only along one direction shows the performance drop compared to the case of exploring all principal directions.

In addition, we also compare our directional attention module with the bidirectional convolutional LSTM module [2] as shown in Table 12. More specifically, we utilize the bidirectional convolutional LSTM in each encoder layer to grasp the spatial correlation instead of the proposed DAM for this experiment. Furthermore, we conducted the additional experiment, which replaces DAM with the spatially adaptive convolution layer [32], and the corresponding result is also shown in Table 13. As can be seen in Tables 12 and 13, the directional attention module (DAM) yields the better performance than other approaches for considering the spatial context regardless of types of backbone architectures.

Table 9 Performance analysis of the proposed method according to the weight to the loss value of each feedback operation on the CULane dataset

# of feedbacks	Weight	
	$\alpha = 1$	$\alpha = 1.1$
	F_1-score	
$T = 1$	74.0	74.2
$T = 2$	74.3	74.3
$T = 3$	74.2	74.4
$T = 4$	74.1	74.5
$T = 5$	74.2	74.4

Table 10 Performance analysis of the proposed method according to the number of scale levels in the encoder

# of scales	TuSimple	CULane	BDD100K	
	Acc.	F_1-score	Acc.	IoU
One level	95.73%	73.8	37.18%	16.53
Two levels	96.01%	74.1	37.75%	16.87
Three levels	96.23%	74.3	37.91%	17.04
Four levels	96.31%	74.4	38.02%	17.09

Note : ERFNet is adopted as the encoder for this experiment

Table 11 Performance analysis of the proposed method according to changes of the recurrent translation process in the directional attention module

Architectures	CULane		
	F_1-score	$Prec.$	$Recall$
Baseline	73.6	74.3	72.9
+ Right direction	73.6	74.2	73.0
+ Down direction	73.8	74.3	73.3
+ Up direction	73.9	74.4	73.4
+ Left direction	73.7	74.1	73.3
Full model (ours)	74.3	74.9	73.8

Note : baseline indicates the directional attention module without the recurrent translation layer in Fig. 2

Table 12 Performance comparison between the proposed directional attention module and the bidirectional convolutional LSTM module

Architectures	CULane		
	F_1-score	$Prec.$	$Recall$
ResNet-34 - BiLSTM [2]	70.6	71.1	70.1
ResNet-34 - Ours	71.9	72.4	71.4
ENet - BiLSTM [2]	71.2	72.6	69.8
ENet - Ours	71.7	72.0	71.4
ERFNet - BiLSTM [2]	73.5	74.4	72.6
ERFNet - Ours	74.3	74.9	73.8

Table 13 Performance comparison of the proposed directional attention module and the spatially adaptive convolution module

Architectures	CULane		
	F_1-score	$Prec.$	$Recall$
ERFNet - SAC [32]	73.9	76.9	71.1
ERFNet - Ours	74.3	74.9	73.8

Note : SAC indicates the spatially adaptive convolution

5 Conclusion

In this paper, a novel method for robust lane detection is proposed. The key idea of the proposed method is to allow for direction-aware features in a global sense, and refine such features based on feedback operations across different scale spaces. To grasp the whole shape of lane areas more accurately, the self-attentive mechanism is applied to directionally weighted features in the proposed DAM. Based on various experimental results on three representative benchmark datasets, i.e., TuSimple, CULane, and BDD100K, we conclude that the proposed method provides reliable lane detection results even with occlusions and other challenging conditions.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Aly M (2008) Real time detection of lane markers in urban streets. IEEE Conf Intell Veh Symp:7–12
2. Azad R, Asadi-Aghbolaghi M, Fathy M, Escalera S (2019) Bi-directional ConvLSTM U-Net with densley connected convolutions. IEEE Conf Int'l Conf Comput Vis Workshop:406–415
3. Bell S, Zitnick CL, Bala K, Girshick R (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. IEEE Conf Int'l Conf Comput Vis Pattern Recognit:2874–2883
4. Borkar A, Hayes M, Smith M (2009) Robust lane detection and tracking with RANSAC and Kalman filter. IEEE Conf Int'l Conf Image Process:3261–3264
5. Borkar A, Hayes M, Smith M (2011) Polar randomized hough transform for lane detection using loose constraints of parallel lines. IEEE Conf Int'l Conf Intell Acoust Speech Signal Process:1037–1040
6. Chen Q, Cheng A, He X, Wang P, Cheng J SpatialFlow: bridging all tasks for panoptic segmentation. IEEE Trans. Circuits Syst. Video Technol. https://doi.org/10.1109/TCSVT.2020.3020257, (Early Access)
7. Chin K-Y, Lin S-F (2015) Lane detection using color-based segmentation. IEEE Conf Intell Veh Symp:706–711
8. Choi H-C, Oh S-Y (2010) Illumination invariant lane color recognition by using road color reference & neural networks. In: Int'l. Joint Conf. Neural Netw, pp 1–5
9. Gao N, Shan Y, Wang Y, Zhao X, Huang K (2021) SSAP: single-shot instance segmentation with affinity pyramid. IEEE Trans Circ Syst Video Technol 31(2):661–673
10. He Y, Wang H, Zhang B (2004) Color-based road detection in urban traffic scenes. IEEE Trans Intell Trans Syst 5(4):309–318
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE Conf Int'l Conf Comput Vis Pattern Recognit:770–778
12. Hu X, Zhu L, Fu C-W, Qin J, Heng P-A (2018) Direction-aware spatial context features for shadow detection. IEEE Conf Int'l Conf Comput Vis Pattern Recognit:7454–7462
13. Hou Y, Ma Z, Liu C, Loy CC (2019) Learning lightweight lane detection cnns by self attention distillation. IEEE Conf Int'l Conf Comput Vis:1013–1021
14. Lee S, Son H, Min K (2010) Implementation of lane detection system using optimized hough transform circuit. IEEE Conf Asia-Pacific Conf Circ Syst:406–409
15. Lee S, Kim J-S, Yoon JS, Shin S, Bailo O, Kim N, Lee T-H, Hong HS, Han S-H, Kweon IS (2017) VPGNet: Vanishing point guided network for lane and road marking detection and recognition. IEEE Conf Int'l Conf Comput Vis:1947–1955
16. Lim KH, Seng KP, Ang L-M, Chin SW (2009) Lane detection and kalman-based linear parabolic lane tracking. IEEE Conf Int'l Conf Intell Hum-Mach Syst Cybern:351–354
17. Neven D, De Brabandere B, Proesmans SM, Van Gool L (2018) Towards end-to-end lane detection: an instance segmentation approach. IEEE Conf Intell Veh Symp:286–291
18. Ozgunalp U, Fan R, Ai X, Dahnoun N (2017) Multiple lane detection algorithm based on novel dense vanishing point estimation. IEEE Trans Intell Trans Syst 18(3):621–632

19. Pan X, Shi J, Luo P, Wang X, Tang X (2018) Spatial cnn for traffic scene understanding. 32nd AAAI Conf Artif Intell:1–8
20. Paszke A, Chaurasis A, Kim S, Culurciello E (2016) ENet: a deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147
21. Paszke A, Gross S, Chintala S, Chanan G, Yang E, Devito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in PyTorch. Adv Neural Inf Process Syst:1–4
22. Qin Z, Wang H, Li X (2020) Ultra fast structure-aware deep lane detection. 15th Eur Conf Comput Vis:276–291
23. Romera E, Alvarez JM, Bergasa LM, Arroyo R (2018) ERFNet: efficient residual factorized ConvNet for real-time semantic segmentation. IEEE Trans Intell Trans Syst 19(1):263–272
24. Satzoda RK, Trivedi MM (2015) On enhancing lane estimation using contextual cues. IEEE Trans Circ Syst Video Technol 25(11):1870–1881
25. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proc. IEEE Int'l. Conf. Comput. Learn. Represent., pp 1–14
26. Su Y, Zhang Y, Lu T, Yang J, Kong H (2019) Vanishing point constrained lane detection with a stereo camera. IEEE Trans Intell Trans Syst 19(8):2739–2744
27. TuSimple (2017) TuSimple Velocity Estimation Challenge in http://github.com/TuSimple/tusimple--benchmark/tre
28. Van Gansbeke W, De Brabandere B, Neven D, Proesmans M, Van Gool L (2019) End-to-end lane detection through differentiable least-squares fitting. IEEE Conf Int'l Conf Comput Vis Workshop 905–913
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst:5998–6008
30. Wang Y, Dahnoun N, Achim A (2012) A novel system for robust lane detection and tracking. Signal Process 92(2):319–334
31. Wu C-B, Wang L-H, Wang K-C (2018) Ultra-low complexity block-based lane detection and departure warning system. IEEE Trans Circ Syst Video Technol 29(2):582–593
32. Xu C, Wu B, Wang Z, Zhan W, Vajda P, Keutzer K, Tomizuka M (2020) SqueezeSegV3: Spatially-adaptive convolution for efficient point-cloud segmentation. 15th Eur Conf Comput Vis:1–19
33. Yang W-J, Cheng Y-T, Chung P-C (2019) Improved lane detection with multilevel features in branch convolutional neural networks. IEEE Access 7:173148–173156
34. Yoo JH, Lee S-W, Park S-K, Kim DH (2017) A robust lane detection method based on vanishing point estimation using the relevance of line segments. IEEE Trans Intell Trans Syst 18(12):3254–3266
35. Yoo S, Lee HS, Myeong H (2020) End-to-end lane marker detection via row-wise classification. IEEE Conf Int'l Conf Comput Vis Pattern Recognit Worshop:1006–1007
36. Yu F, Chen H, Wang X, Xian W, Chen Y, Liu F, Madhavan V, Darrell T (2020) BDD100K: a diverse driving dataset for heterogeneous multitask learning. IEEE Conf Int'l Conf Comput Vis Pattern Recognit:2636–2645
37. Yuan J, Tang S, pan X, Zhang H (2014) A robust vanishing point estimation method for lane detection. Chin Control Conf:4887–4892
38. Zhang J, Xu Y, Ni B, Duan Z (2018) Geometric constrained joint lane segmentation and lane boundary detection. 14th Eur Conf Comput Vis:486–502
39. Zhang X, Li H, Meng F, Song Z, Xu L Segmenting beyond the bounding box for instance segmentation. IEEE Trans. Circuits Syst. Video Technol. https://doi.org/10.1109/TCSVT.2021.3063377, (Early Access)
40. Zhao K, Meuter M, Nunn C, Müller D, Müller-Schneiders S, Pauli J (2012) A novel multi-lane detection and tracking system. IEEE Conf Intell Veh Symp:1084–1089
41. Zhou S, Jiang Y, Xi J, Gong J, Xiong G, Chen H (2010) A novel lane detection based on geometrical model and gabor filter. IEEE Conf Intell Veh Symp:59–64

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.