Attention-based Row Selecting Networks for Lane Detection

Man Jiang
Guizhou Minzu University

Qian Zhang (✉ 331394262@qq.com)
Guizhou Minzu University

Yuhang Zhang
Guizhou Minzu University

Jiangtao Su
Guizhou Minzu University

Research Article

Keywords: Lane detection, Attention mechanism, Group normalization, Deep learning

Posted Date: August 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1799743/v2

Additional Declarations: No competing interests reported.

Attention-based Row Selecting Networks for Lane Detection

Man Jiang[1,2], Qian Zhang[1,3*], Yuhang Zhang[1,2] and Jiangtao Su[1,2]

[1*]School of Data Science and Information Engineering, Guizhou Minzu University, Guiyang, 550025, Guizhou, China.
[2]Key Laboratory of Pattern Recognition and Intelligent Systems of Guizhou,Guiyang, 550025, Guizhou, China.
[3]Academic Affairs Office, Guizhou Minzu University, Guiyang, 550025, Guizhou, China.

*Corresponding author(s). E-mail(s): 331394262@qq.com;
Contributing authors: 541726282@qq.com; 953307200@qq.com; 1902563708@qq.com;

Abstract

Most mainstream methods mainly regard lane detection as a pixel-by-pixel segmentation task, resulting in high computational cost and time-consuming, and the accuracy is influenced by severe occlusion and extreme lighting conditions. To tackle these issues, we propose a novel Attention-based Row Selecting Networks(ARS-Net), which utilizes the row selecting method based on global features to detect lanes, greatly improves the detection speed. At the same time, channel and spatial attention mechanisms are integrated into ResNet as the backbone to focus on important features and suppress unimportant ones, so as to adjust feature weights and reduce information loss. Besides, group normalization is employed to replace batch normalization, which enhances the stability of accuracy. We carry out immense amounts of experiments on two international public datasets TuSimple and CULane, the experimental results show that our method achieves the state-of-the-art performance in both accuracy and speed and significantly outperforms other methods for real-time and efficient lane detection in real-world applications.

Keywords: Lane detection, Attention mechanism, Group normalization, Deep learning

1 Introduction

With the improvement of living standards, automobiles have become a very common transportation means, while the incidence of traffic accidents is also increasing year by year. According to relevant surveys, about half of traffic accidents are related to lane deviation. Research shows that if the driver's reaction time is improved by 0.5s, at least 60% of rear-end crashes, 50% of cross collisions and 30% of head-on collisions can be avoided[1]. Therefore, improving the lane detection technology and the Intelligent Driver Assistance Systems(IDAS) that provides assistance to drivers is of great significance to reduce the incidence of traffic accidents and ensure the safety of driving.

As a fundamental task of autonomous driving, lane detection based on computer vision has been widely applied in Adaptive Cruise Control(ACC), Blind Spot Monitoring(BSM), Lane

Departure Warning(LDW) and other driver assistance systems[2]. In real driving scenarios, the road environment is complex and changeable, affected by weather, light conditions, lane wear, road shadow, and lane occlusion by passing vehicles, many lane detection methods have the problems of high false and missed detection rate and low accuracy, as well as large amounts of computation and long time consuming.

To deal with these issues, many scholars have carried out extensive research. At present, there are two mainstream lane detection methods, which are traditional methods and deep learning methods. Traditional methods exploit hand-designed features to detect lane, such as grayscale[3], color[4], texture[5], edge gradient[6], and visual vanishing point[7], which are sensitive to environmental changes such as light and weather. When the driving environment changes significantly, the effect of lane detection becomes worse.

With the development of deep learning, deep neural networks have shown superiority in object detection due to their great representation and learning ability. Compared with traditional methods, feature extraction based on deep learning can gain richer information, and have been widely used in lane detection. DeepLanes[8] utilized a deep convolutional neural network for feature extraction, and performed pixel-wise classification of lane and background to achieve end-to-end lane detection. SegNet[9] proposed an improved segmentation network, which adopts convolution and pooling operations to extract lane features, and exploits connected domain constraints to achieve accurate segmentation of binary lane images. However, its structure is complex and real-time performance is poor. LaneNet[10] proposed an instance segmentation method, which can cluster each lane in training stage rather than the post-processing stage, greatly improving the running speed. LSTR[11] developed a transformer-based network for end-to-end lane detection with a self-attention mechanism to capture slender structure of lanes and rich global context information, achieving high accuracy and speed.

The above methods all detect lane pixel-by-pixel, requiring high computing cost. Qin et al.[12] regarded lane detection as a row-based selecting problem using global features. Instead of segmenting every pixel of image, the location of lane was selected at predefined rows of the image, which significantly reduces the computational cost. It achieved ultra fast speed and solved the no-visual-clue problem. Inspired by it, we proposes an efficient new method for lane detection. The ResNet integrated with the channel and spatial attention mechanisms is used as backbone to extract the global features of the image. Then, the row-based selecting method based on global features is used to select the lane position to achieve real-time and accurate detection. Our main contribution can be summarized in four parts:

- We propose a new Attention-based Row Selecting Networks(ARS-Net) for lane detection, which integrates the Convolutional Block Attention Module(CBAM) into ResNet to adjust feature weights, so as to reduce information loss by paying attention to important features and suppress unnecessary ones.
- Group normalization is used to replace batch normalization, which improves the stability of accuracy.
- We demonstrate the effectiveness of our model by conducting extensive ablation studies.
- We experiment on two public datasets TuSimple and CULane and compare with other methods, verifying that the proposed method achieves the state-of-the-art performance in terms of both accuracy and speed.

2 Related Work

Attention mechanism. Attention mechanism tells network where to focus, it enhances the representation power of network by focusing on important features and suppressing unimportant ones. In recent years, some scholars have tried to incorporate attention mechanism into CNN to improve its performance. Hu et al.[13] introduced a Squeeze-and-Excitation block(SE) to focus on the relationship between channels and exploited global average pooling features to selectively emphasize informative features and suppress less useful ones, thus improving the representation power of the network. However, Woo et al.[14] validated that adopting both spatial attention and channel attention is superior to just using channel attention. A new Convolutional Block Attention Module(CBAM) is proposed, which sequentially applies channel and spatial attention module to extract features and learn 'what' and 'where' to pay attention to by mixing channel and spatial information together. In this paper, CBAM is integrated into CNN to adjust the feature weights of input images to enhance classification capability.

Normalization. Normalization has been widely used in deep networks as normalizing the input data makes training faster. AlexNet[15] proposed Local Response Normalization(LRN), which normalized the convolution results of adjacent depths at the same spatial position and slightly improve the generalization ability of the model. Batch Normalization(BN)[16] normalizes the features by calculating mean and variance within a batch, which enables very deep networks to converge and benefits generalization. However, BN is heavily dependent on batch size, too small batch size will result in performance degradation. In addition, the batch dimension may be different during training and testing, which will lead to inconsistency problem.

Several normalization methods have been proposed to avoid using the batch dimension. Layer Normalization(LN)[17] normalizes each feature map along the channel dimension. Instance Normalization(IN)[18] normalizes within each channel for each image. However, the accuracy of LN and IN is not so good as BN in many visual recognition tasks. Wu et al.[19] proposed Group Normalization(GN), it divides the channels into groups and normalizes within each group. GN is comparably good with BN and its accuracy is more stable than BN in a wider batch range as its calculation is independent of batch size. Hence, we adopt GN to normalize.

3 Method

The overall architecture of proposed Attention-based Row Selecting Networks(ARS-Net) is shown in Figure 1. The ResNet integrates with the Convolutional Block Attention Module(CBAM) is regarded as the backbone for the extraction of global image features, then the row-based selecting method based on global features is utilized to classify the predefined row anchors and predict the lane position. In addition, an auxiliary segmentation branch is added during training, which are composed of multiple convolution, group normalization, and ReLU activation layers, to aggregate global and local features. Auxiliary segmentation is only worked during training and is removed during testing. In this way, the running speed will not be affected while the multi-scale features are well aggregated for lane segmentation.

3.1 CBAM-ResNet

In order to pay more attention to important features of images, we add channel and spatial

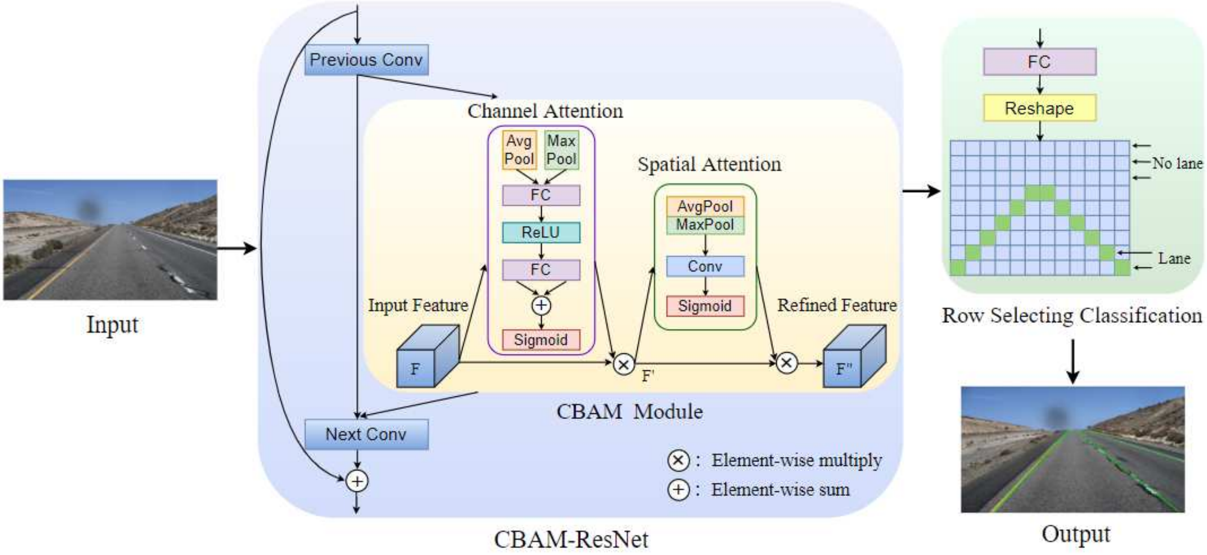


Fig. 1 Overall Architecture.The feature extractor is illustrated in the blue box, the classification prediction of lanes is shown in the green box. The group classification is conducted on each row anchor.

attention mechanisms(CBAM) between each two convolution layers to adjust feature weights.

For the channel attention mechanism, first input feature map F, perform both global average-pooling and max-pooling operations based on width and height to aggregate spatial information of the feature map, generating F_{avg} and F_{max}. Then, both F_{avg} and F_{max} are forwarded to two shared fully-connected(FC) layers, the first FC layer compresses C channels into C/r channels to reduce parameter overhead, r is the reduction ratio, followed by ReLU activation function, the second FC layer increases the dimension to return to C channels. The output features are merged by element-wise summation, and the final channel attention feature F_C is generated by the sigmoid activation function. At last perform element-wise multiplication of the channel attention feature F_C and the input feature F to generate F' as the input feature of the Spatial attention module. In short, the channel attention mechanism learns "what" to pay attention to, which is computed by:

$$\begin{aligned} F_C &= \sigma\left(\mathrm{FC}_2\left(\mathrm{FC}_1(\mathrm{AvgPool}(\mathrm{F}))\right)\right. \\ &\quad \left.+\,\mathrm{FC}_2\left(\mathrm{FC}_1(\mathrm{MaxPool}(\mathrm{F}))\right)\right) \\ &= \sigma\left(W_2\left(W_1\left(F_{\text{avg}}\right)\right)+W_2\left(W_1\left(F_{\max}\right)\right)\right) \end{aligned} \tag{1}$$

where σ refers to the sigmoid function, W_1 and W_2 denotes the FC shared weights.

For the spatial attention mechanism, input feature F', first perform channel-based global average-pooling and max-pooling, and then concatenate the two generated features F'_{avg} and F'_{max}. Then reduce the dimension to 1 channel by a standard convolution layer. Finally, the spatial attention feature F_S is obtained by a sigmoid function. And the final refined output feature F'' is calculated by element-wise multiplication of the spatial attention feature F_S and the input feature F'. In a word, the spatial attention mechanism learns "where" to pay attention, which can be expressed as:

$$\begin{aligned} F_S &= \sigma\left(f_{7\times 7}\left(\left[\mathrm{AvgPool}\left(F'\right);\ \mathrm{MaxPool}\left(F'\right)\right]\right)\right. \\ &= \sigma\left(f_{7\times 7}\left[F'_{\text{avg}}\ ;\ F'_{\max}\ \right]\right)\right) \end{aligned} \tag{2}$$

where $f_{7\times 7}$ represents a convolution operation with the filter size of 7×7.

The overall CBAM attention process can be summarized as:

$$\begin{aligned} F' &= F \otimes F_C \\ F'' &= F' \otimes F_S \end{aligned} \tag{3}$$

where $\otimes$ denotes element-wise multiplication.

By plugging CBAM module after each non-linear layer following convolution of ResNet, the CBAM-ResNet can learn which information to emphasize and which to suppress, help to adjust the feature weights of the input image and capture non-local contextual information to compensate for the missing details of lanes, efficiently improve the performance of lane detection.

3.2 Row-based selecting method for lane detection

Like[12], we exploit the row-based selecting method for lane detection. Firstly is gridding, the image is divided into several rows and judged whether each row anchor has lanes according to the global features extracted by CBAM-ResNet. Predefined row anchors are those rows with lanes, each row is further divided into multiple cells, and lane detection is to select specific cells as lane positions on predefined row anchors.

Suppose the input image size is $H\times W$, the maximum number of lanes is C. Let h represent the number of predefined row anchors, and w be the number of grid cells. It is obviously that $h << H$ and $w << W$, that is, the number of predefined row anchors is far less than the height of image and the number of grid cells is far less than the width of image. In this way, the segmentation method needs to carry out $H\times W$ classifications that are $(C+1)$-dimensional, while ours only needs to conduct $C\times h$ classifications that are $(w+1)$-dimensional. The extra dimension indicates the absence of lane. $C\times h\times(w+1)$ is much less than $H\times W\times(C+1)$, hence the amount of computation is drastically reduced.

Compared with the traditional pixel-by-pixel segmentation method, our method which selected the lane position based on the predefined row anchors whose number and gridding size are far less than the size of the original image greatly reduces the calculational cost, so the detection speed is extremely fast. At the same time, our method uses global features as input, having a larger receptive field than segmentation methods, which is beneficial to solve the problem of no visual clue of lanes by supplementing contextual information for the missing details of lanes under occlusion or severe weather and light.

3.3 Group Normalization

Group normalization(GN) is to normalize each group. It divides the channels into multiple groups, and normalize the features by calculating the mean and variance within each group. The calculation formula for GN is:

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \varepsilon}} \times \gamma + \beta \tag{4}$$

where $E(x)$ and $Var(x)$ denote the mean and variance, ε is a small constant, γ and β are trainable parameters representing scale and shift.

In the case of two-dimensional images, the feature dimension is $[N, C, H, W]$, where N is the batch axis, C is the channel axis, and H and W are the spatial height and width axes[19]. Batch Normalization(BN) normalizes along batch dimension, the normalized dimension is $[N, H, W]$. Layer Normalization(LN) and Instance Normalization(IN) avoids the batch dimension and the normalized dimensions are $[C, H, W]$ and $[H, W]$ respectively.

GN combines the advantages of LN and IN, it reshapes the feature dimension from $[N, C, H, W]$ to $[N, G, C/G, H, W]$ and the normalized dimension is $[C/G, H, W]$. Here G is the number of groups, which is a predefined hyper-parameter. C/G is the number of channels per group.

GN's computation is independent of batch sizes, which avoids inconsistency problem of batch dimension using BN during training and testing, and its accuracy is comparable to that of BN which is better than LN and CN in many visual recognition tasks while more stable than BN in a wide range of batch sizes, thus we exploit GN to replace BN for feature normalization.

4 Experiments

4.1 Datasets

We experiment on two widely-used public lane detection datasets, TuSimple[20] and CULane[21]. The TuSimple dataset is collected on the highway in the daytime, consisting of 6408 video clips, each clip has 20 frames and the last frame is annotated. It is divided into training set of 3268 frames, validation set of 358 frames and test set of 2782 frames. The image resolution is 1280×720 pixels and marking up to 5 lanes. The CULane dataset has a total of 133235 frames, which are divided into 88880 frames for training, 9675 for validation and 34680 for testing. It consists of nine different scenarios: normal, crowded, night, no-line, shadow, arrow, dazzle light, curve, and crossroad, with a size of 1640×590 and marking up to 4 lanes.

4.2 Evaluation metrics

The evaluation metrics of TuSimple dataset are accuracy, false positive rate (FPR) and false negative rate (FNR). Generally speaking, the higher the accuracy, the lower the FPR and FNR, the better the lane detection effect. Accuracy is calculated by:

$$Accuracy = \sum_{im} \frac{TP_{im}}{GT_{im}} \tag{5}$$

where TP_{im} is the number of true prediction points of the image, and GT_{im} is the number of ground truth points[11]. FPR and FNR are formulated as follows:

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

$$FNR = \frac{FN}{TP + FN} \tag{7}$$

where TP is the true positive, FP is the false positive, TN is the true negative and FN is the false negative. FPR represents the proportion of all negative samples that are wrongly predicted to be lane. FNR is the proportion of all real lanes that are not predicted.

F1-measure is taken as the evaluation metric of CULane dataset and calculated by:

$$Fl - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

here,

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TN}{TP + FN} \tag{10}$$

Precision represents the proportion of true positive in all predicted positive samples. Recall means the proportion of all positive samples that are correctly predicted.

4.3 Implementation Details

Experimental platform. Operating system Windows 10×64, processor Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz 3.70 GHz, graphics card NVIDIA GeForce RTX 3080Ti, all experiments are run based on the deep learning framework Pytorch.

Parameter setting. The row anchors defined by TuSimple dataset range from 160 to 710, and the counterpart of CULane dataset range from 260 to 530. Set the number of grid cells to 100 on TuSimple and 200 on CULane. The input images are resized to 288×800 and the batch size is 32. The network is optimized by Adam optimizer with the learning rate of 4×10^{-4}, the momentum of 0.9, the weight decay of 1×10^{-4}. The iteration epoch is set to 100 for TuSimple dataset and 50 for CULane dataset.

Data augmentation. To prevent over-fitting, the raw data are augmented by rotating, cropping, horizontal and vertical shift. Furthermore, the lanes are extended to the boundary of the image to preserve the lane structure.

4.4 Ablation Study

In order to verify the effectiveness of the proposed modules, we conduct ablation experiments based on ResNet-18 backbone with different module combinations. The experiments are carried out under the same training settings and the quantitative results on TuSimple and CULane datasets are shown in Table 1 and Table 2.

Table 1 shows the experimental results of the proposed modules on TuSimple dataset with ResNet-18 backbone. It can be seen that inserting channel attention mechanism into the backbone can improve the detection effect, the accuracy is increased, and the false positive rate and false negative rate are decreased. What's more, adding the spatial attention mechanism further gains performance improvement, so it is better to utilize both channel and spatial attention mechanism. For normalization, it can be seen that the detection performance is also enhanced by changing the batch normalization to group normalization from the original ResNet18 backbone. Ultimately we achieve the best performance by employing both channel and spatial attention and group normalization, verifying the effectiveness of the proposed modules.

Table 2 shows the F1-measure experiment on CuLane dataset. We can see that the addition

Table 1 Experiments of the proposed modules on Tusimple dataset with ResNet-18 backbone.

Backbone	Normalization	Accuracy(%)	FPR	FNR
ResNet18[12]	BN	95.81	0.1905	0.0392
ResNet18+channel	BN	95.87	0.1886	0.0367
ResNet18+channel+spatial	BN	95.91	0.1881	0.0365
ResNet18	GN	95.88	0.1882	0.0366
ResNet18+channel+spatial	BN	**95.94**	**0.1875**	**0.0356**

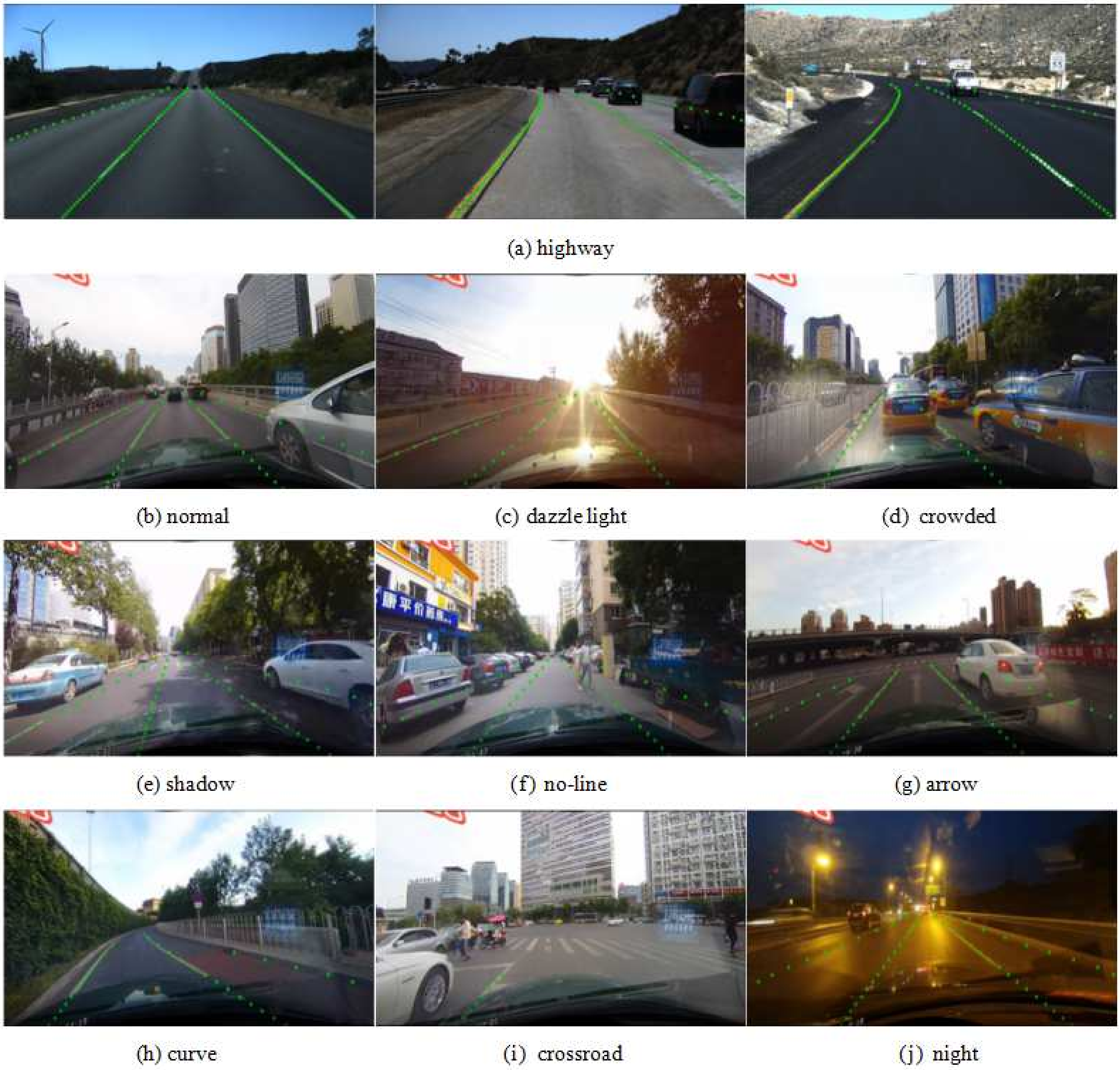
Table 2 Experimental results of F1-measure on CULane dataset based on ResNet-18 backbone with different module combinations.

Backbone	Norm	Normal	Crowd	Night	No-line	Shadow	Arrow	Dazzle light	Curve	Total
ResNet18[12]	BN	87.7	66.0	62.1	40.2	62.8	81.0	58.4	57.9	68.4
ResNet18+channel	BN	88.9	67.7	63.7	40.1	61.1	81.9	56.3	57.6	69.4
ResNet18+channel+spatial	BN	89.3	68.1	64.3	40.6	65.4	84.1	58.6	58.5	70.3
ResNet18	GN	89.2	**68.2**	64.0	40.7	63.8	83.7	60.4	58.1	70.1
ResNet18+channel+spatial	GN	**89.4**	68.1	**64.5**	**40.9**	**67.5**	**85.0**	**60.7**	**59.3**	**70.8**

of channel attention mechanism can enhance the detection performance in most scenarios, except for shadow and dazzle light scenes, which may be due to the channel attention is not robust enough for the change of light conditions. Besides, both adding both channel and spatial attention and exploiting group normalization can boost the F1-measure of all scenarios. The same as TuSimple dataset, adopting both channel and spatial attention and changing batch normalization to group normalization can achieve the best performance. Compared with the original method, our method significantly enhances the performance for lane detection, which corroborates its effectiveness.

4.5 Results and Comparisons

The visualization of our lane detection results are given in Figure 2. It can be seen that our method can accurately identify the position of lanes in various challengeable scenarios such as curve, dazzle light, crowded, shadow, no-line, night and so on. Moreover, the lanes occluded by passing vehicles and pedestrians can also be well detected (as shown in Figure 2(d) and (f)).



(a) highway

(b) normal

(c) dazzle light

(d) crowded

(e) shadow

(f) no-line

(g) arrow

(h) curve

(i) crossroad

(j) night

Fig. 2 Visualization of lane detection results. The first row are the results on TuSimple dataset, and the rest are on CULane dataset.

Figure 3 and Figure 4 shows the qualitative comparison of our method and UFS[12] with the same ResNet-18 backbone on TuSimple and CULane datasets respectively. It is observed that the detection performance of our method is significantly superior to that of UFS, which can primely capture the structure of lanes and greatly improve the phenomenon of false detection (Figure 4 row 1, column 2 and 3) and missed detection (Figure 4 row 3, column 1) of UFS. We attribute this to the fact that the attention mechanism we added is capable of capturing non-local information, supplementing contextual information for the missing details, which helps to catch the structure of lane. Subsequent quantitative comparison further confirms the effectiveness of our method.

Table 3 shows the quantitative comparison of detection accuracy and speed between our method and other 7 state-of-the-art lane detection methods on Tusimple dataset, including SCNN[21], Seg[22], LaneNet[23], SAD[24], FastDraw[25], PolyLaneNet[26] and UFS[12]. In these experiments, ResNet-18 and ResNet-34 are used as backbone.

It can be seen that our method achieves comparable performance with state-of-the-art methods while our speed is the fastest, with an average runtime of 2.4ms, reaching 414.3 FPS, which is 56 times faster than that of the SCNN. Our ARS-Res18 outperforms the Seg-Res18 by 3.25% accuracy and runs 10.5 times faster, as well as in the Res34 case, which demonstrates that our row-based selecting method superior to the segmentation method. Compared with the state-of-the-art method ENet-SAD, ours is only 0.7% lower in accuracy but runs 5.6 times faster. Compared to UFS with the same backbone, our method can

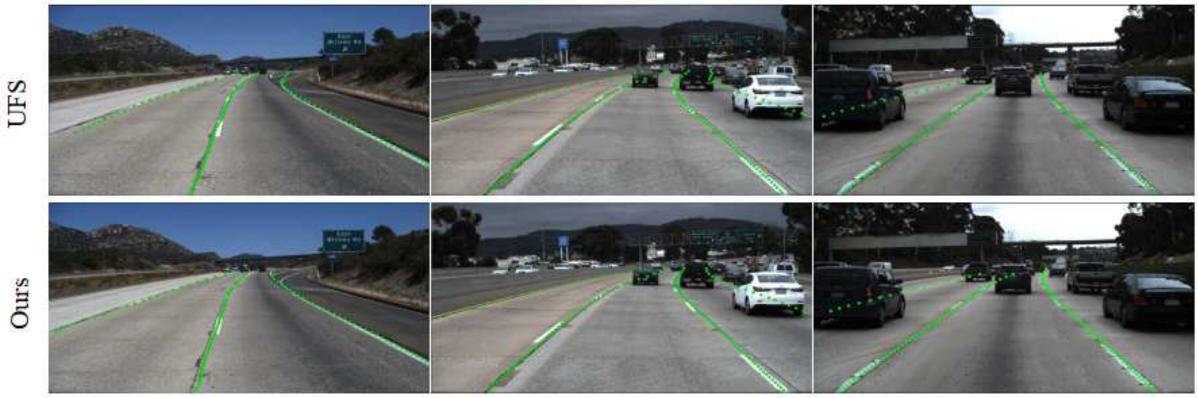


Fig. 3 Qualitative comparison with UFS on TuSimple. The first row visualize the predicted results by UFS, and the rest visualize our predictions.

Table 3 Quantitative comparison with state-of-the-art methods on Tusimple. The runtime is the average time for 100 runs, and the calculation of runtime multiple is based on the slowest method SCNN.

Method	Accuracy(%)	Runtime(ms)	FPS	Multiple
SCNN[21]	96.53	133.5	7.4	1.0x
Seg-Res34[22]	92.84	50.5	19.8	2.6x
Seg-Res18[22]	92.69	25.3	39.5	5.3x
LaneNet[23]	96.38	19.0	52.6	7.0x
ENet-SAD[24]	**96.64**	13.4	74.6	10.0x
FastDraw[25]	95.20	11.1	90	12.1x
PolyLaneNet[26]	93.36	8.7	115	15.5x
UFS-Res34[12]	95.87	5.9	169.4	22.6x
UFS-Res18[12]	95.81	3.2	312.5	41.7x
ARS-Res34(Ours)	96.02	4.6	217.4	29.4x
ARS-Res18(Ours)	95.94	**2.4**	**414.3**	**56.0x**

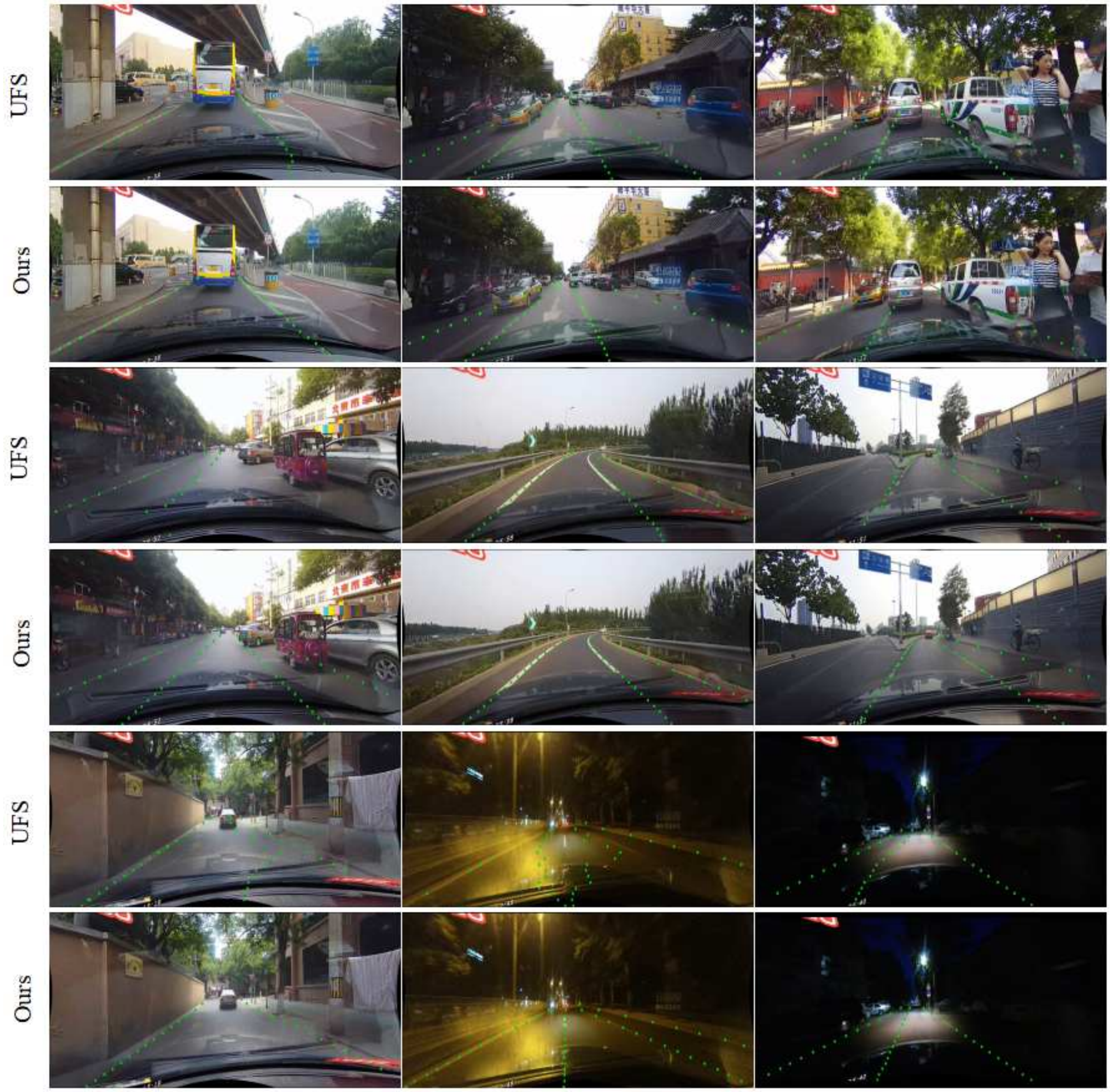


Fig. 4 Qualitative comparison with UFS on CULane. The odd-numbered rows visualize the predicted results by UFS, and the even-numbered rows visualize our predictions.

achieve both better performance and faster speed, which corroborates the effectiveness of our attention mechanism and group normalization module.

For CULane dataset, the comparisons of F1-measure and runtime are shown in Table 4. It is observed that our method achieves the best performance in terms of both accuracy and speed. The F1-measure of our ARS-Res34 is the highest in most scenarios, with the total F1-measure of 72.1, an improvement of 1.4 compared to that of SAD-Res34 and UFS-Res34 with the same backbone. Meanwhile, our ARS-Res18 gains the fastest detection speed among all methods, reaching 409.3 FPS, which is 55.6 times faster than that of the second-best SCNN. In summary, our method significantly outperforms the state-of-the-art methods with the fastest detection speed and much higher accuracy, hence has enormous mobile deployment capability.

Table 4 Quantitative comparison of F1-measure and runtime on CULane.

Scenario	SCNN [21]	SAD-Res18 [24]	SAD-Res34 [24]	UFS-Res18 [12]	UFS-Res34 [12]	ARS-Res18 (Ours)	ARS-Res34 (Ours)
Normal	**90.6**	89.8	89.9	87.7	89.4	89.4	90.0
Crowded	69.7	68.1	68.5	66.0	69.3	68.1	**69.8**
Night	66.1	64.2	64.6	62.1	65.1	64.5	**66.5**
No-line	**43.4**	42.5	42.2	40.2	41.1	40.9	42.7
Shadow	66.9	67.5	67.7	62.8	62.6	67.5	**69.9**
Arrow	84.1	83.9	83.8	81.0	84.9	85.0	**85.4**
Dazzle light	58.5	59.8	59.9	58.4	58.6	60.7	**62.2**
Curve	64.4	65.5	**66.0**	57.9	58.7	59.3	63.2
Total	71.6	70.5	70.7	68.4	70.7	70.8	**72.1**
Runtime(ms)	133.5	25.3	50.5	3.1	5.7	**2.4**	4.7
FPS	7.5	39.5	19.8	322.5	175.4	**409.3**	213.4
Multiple	1.0x	5.3x	2.6x	43.0x	23.4x	**55.6x**	28.5x

5 Conclusion

In this work, a novel lane detection method, Attention-based Row Selecting Networks(ARS-Net), is proposed, in which channel and spatial attention mechanisms are integrated into ResNet as backbone to adjust feature weights and reduce information loss. The position of lane is identified by row selecting method based on global feature, which vastly improves the detection speed. And group normalization is exploited instead of batch normalization to enhance the stability of accuracy. We verify the effectiveness of the proposed modules by conducting extensive ablation experiments on two public datasets Tusimple and CULane. Compared with other methods, our method achieves state-of-the-art performance in terms of accuracy and speed, which is capable of practical application.

Declarations

Ethical Approval. Not applicable. (This article does not contain any studies with human participants or animals performed by any of the authors.)

Competing Interests. The authors have no competing interests that are relevant to the content of this paper.

Authors' contributions. Man Jiang: Writing, Experiment, Analysing and Finalizing; Qian Zhang: Reviewing and Supervision; YuHang Zhang: Writing, Editing and Experiment; Jiangtao Su: Editing and Simulation. All authors read and approved the final manuscript.

Funding. This work is supported in part by the Natural Science Foundation of China under Grant 61802082, 61263034 and 61762020, in part by the Department of Education of Guizhou Province under Grant QIAN JIAO HE KY[2018]018.

Availability of data and materials. The datasets and code generated and analysed during the current study are available from the corresponding author on reasonable request.

References

1. Gao S, Dong T, Wang C (2020) Simulation and Optimization of Lane Detection System Based on Machine Learning. Computer Simulation 37(02), 140-143
2. Narote S P, Bhujbal P N, Narote A S, et al (2018) A review of recent advances in lane detection and departure warning system. Pattern Recognition 73: 216-234
3. Parajuli A, Celenk M, Riley H B (2013) Robust lane detection in shadows and low illumination conditions using local gradient features. Open Journal of Applied Sciences 3(01): 68
4. Somawirata I K, Utaminingrum F (2016) Road detection based on the color space and cluster connecting. In: 2016 IEEE International Conference on Signal and Image Processing (ICSIP), pp 118-122

5. Li Z Q, Ma H M, Liu Z Y (2016) Road lane detection with Gabor filters. In: 2016 International Conference on Information System and Artificial Intelligence (ISAI), pp 436-440

6. Mu C, Ma X (2014) Lane detection based on object segmentation and piecewise fitting. TELKOMNIKA Indonesian Journal of Electrical Engineering 12(5): 3491-3500

7. Hou C, Hou J, Yu C (2016) An efficient lane markings detection and tracking method based on vanishing point constraints. In: 2016 35th Chinese Control Conference (CCC), pp 6999-7004

8. Gurghian A, Koduri T, Bailur S V, et al (2018) Deeplanes: End-to-end lane position estimation using deep neural networksa. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 38-45

9. Deng T, Wang L, Yang Q, et al (2020) Lane line detection method based on improved SegNet algorithm. Science Technology and Engineering 20(36): 14988-14993

10. Wang Z, Ren W, Qiu Q (2018) LaneNet: Real-time lane detection networks for autonomous driving. arXiv preprint arXiv: 1807.01726

11. LIU R, Yuan Z, Liu T, et al (2021) End-to-end lane shape prediction with transformers. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp 3694-3702

12. Qin Z, Wang H, Li X (2020) Ultra fast structure-aware deep lane detection. In: European Conference on Computer Vision. Springer, Cham, pp 276-291

13. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 7132-7141.

14. Woo S, Park J, Lee J Y, et al (2018) CBAM: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3-19

15. Krizhevsky A, Sutskever I, Hinton G E (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25

16. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448-456

17. Ba J L, Kiros J R, Hinton G E (2016) Layer normalization. arXiv preprint arXiv:1607.06450

18. Ulyanov D, Vedaldi A, Lempitsky V (2016) Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022

19. Wu Y, He K (2018) Group normalization. In: Proceedings of the European conference on computer vision (ECCV), pp 3-19

20. Tusimple benchmark (2017) https://github.com/TuSimple/tusimple-benchmark

21. Pan X, Shi J, Luo P, et al (2018) Spatial as deep: Spatial cnn for traffic scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 7276– 7283

22. Chen L C, Papandreou G, Kokkinos I, et al (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40(4): 834-848

23. Neven D, De Brabandere B, Georgoulis S, et al. Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE, pp 286-291

24. Hou Y, Ma Z, Liu C, et al (2019) Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1013-1021

25. Philion J (2019) 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, pp 11582-11591

26. Tabelini L, Berriel R, Paixao T M, et al (2021) Polylanenet: Lane estimation via deep polynomial regression. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, pp 6150-6156