Hindawi
Mobile Information Systems
Volume 2023, Article ID 2075022, 9 pages
https://doi.org/10.1155/2023/2075022

Research Article

Lane Marker Detection Based on Multihead Self-Attention

Fan Shengli, Zhang Yuzhi, and Bi Xiaohui

Department of Automotive Engineering, Hebei Vocational University of Industry and Technology, Hongqi Avenue 626, Shijiazhuang 050091, China

Correspondence should be addressed to Fan Shengli; hbgyyb@hbcit.edu.cn

Received 13 December 2022; Revised 6 January 2023; Accepted 14 January 2023; Published 3 February 2023

Academic Editor: Floriano Scioscia

Lane mark detection is an important task for autonomous driving. Many researchers have proposed many models. But the driving environment is much more complex, especially for some challenging scenarios, such as vehicle occlusion, severe mark degradation, heavy shadow, and so on. It is difficult to detect lane mark in a limited local receptive field under the above scenarios. For that reason, we propose a lane mark detection network based on multihead self-attention. It can find spatial relationships among lane mark points in the global viewpoint and enlarge its feature map's receptive field equally. For further extracting global and contextual features, it fuses global information and local information together to predict classification and location regression. Finally, it can promote accuracy of lane mark detection greatly especially in challenging scenarios. In the TuSimple benchmark, its accuracy is 95.76% overwhelming all other methods, and its FPS is 170.2, which is the second-highest one. In CULane benchmark its F1 achieves 75.55% and FPS reaches 170.5. Both of them are the highest compared to other methods. Our proposed model establishes a new state-of-the-art among real-time methods.

1. Introduction

Lane detection [1, 2] based on vision sensors is one of the core technologies in the auto-driving field. Currently, it is not only an important foundation for lane departure warning and lane keeping functions but also a key technology to accomplish ADAS (advanced driving aided system) [3, 4]. However, there are so many sorts of lanes in the realization world. For example, there are solid, broken, dash, merging, and splitting lanes. Lane patterns are diverse. Besides that, there are some challenging driving scenarios, including those involving heavy shadows, severe vehicle occlusion, and severe road mark degradation. Even so, there are some corner cases such as merging and splitting. In an urban environment, lanes are susceptible to illumination, load wear and tear, occlusion, etc. It is more challenging and makes a higher claim to algorithm generalization and robustness.

To resolve those existing problems, many researchers put forward some different technical solutions. In traditional computer vision, it heavily depends on some assumptions, such as that lanes and boundaries are continuous and parallel [5]. Also, it utilizes edge detection operators, a histogram, prior knowledge, and recognition to extract lane candidate points. Finally, it takes advantage of line fitting or the Hough [6–9] transformation to obtain the lane line parameters. More recently, the development of CNN semantic segmentation [10–16] or instance segmentation [17–21] is paid most attention. It extracts spatial or structural information between pixels or from slice to slice in the process of lane detection [22–26]. Although it can resolve some challenging scenarios like vehicle occlusion, severe road mark degradation, and heavy shadows, its huge computation cost and much slower speed hinder its real-time application and performance, as shown in Figure 1. Consequently, recurrent neural networks, long short-term memory, gated recurrent neural networks, and attention mechanisms have been firmly established. They do well in coping with time series signal processing and sequence modeling. Especially for lane line occlusion, it can extract textual or semantic information from continuous frames.

In this work, we present a lane mark detection network based on multihead self-attention [27]. It is a lightweight model and is applied in real-time application. Its accuracy is

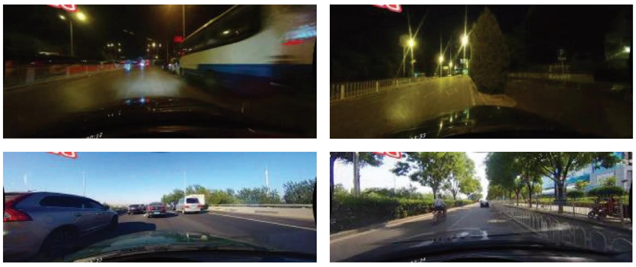

Figure 1: Illustration of challenging scenarios in lane mark detection. Most challenging scenarios are vehicle occlusion, huge shadows, and severe road mark degradation which result from lane detection difficulties.

much better than most state-of-the-art models. TuSimple and CULane are used as our benchmarks to evaluate our experimental results. This paper has three extensions, as follows:

(i) A lane mark detection network based on anchors and multiheader self-attention: we propose a new network architecture combining row anchors with multiheader self-attention. It promotes accuracy a lot compared with that in [17, 28–32].

(ii) Multiheader self-attention mechanism: we propose a multihead self-attention method to extract global information, which further improves the performance.

(iii) Presentations and experiments: two datasets are collected for performance evaluation. One is TuSimple dataset, and the other one is CULane dataset. These two benchmarks are utilized for quantitative evaluation for different scenarios, such as city lanes and rural lanes, in day and night conditions. It can promote the research and development of autonomic driving.

2. Related Work

In the past two decades, researchers have made great efforts on lane detection technology. Especially when DCNN, LSTM, and Attention emerge, it brings a new viewpoint to lane detection methods. Totally, these methods are sorted to some categories such as traditional methods, segmentation network, anchor-based methods, and attention-based methods. In this section, we briefly summarize each category.

2.1. Traditional Computer Vision-Based Lane Detection. Generally speaking, traditional computer vision methods are mainly concerned with gray images, edge detection operators, and ROI in order to detect lane edges. Generally, it divides lane detection into two stages. One stage is lane edge searching and detection. During lane edge detection processing, it takes the IPM transformation, Sobel operator, Gaussian filter, steerable filter [33], and Gabor filter [34] with kernels in different directions, gradient, color, and texture. The other stage is lane fitting. So, many methods are extensively exploited to fit lane line; the input is a gray image, not the original RGB image. It brings about multi-preprocessing methods such as template matching [35], Hough transformation, polar randomized HT [36], curve-line fitting, Catmull-Rom spline [37], B-snake [38], and so on.

2.2. Lane Detection Based on Segmentation. Global information, local information, textual information, and semantic information are very important for lane detection, especially in vehicle occlusion scenarios. The segmentation network intensifies communication among pixels in a larger receptive field. The main research directions are as follows:

(1) Pixel-wise segmentation: the authors in [39] propose atrous convolution and bilinear interpolation to acquire a larger receptive field in order to get much higher classification accuracy. It utilizes atrous spatial pyramid pooling with different sampling rates to aggregate multiscale feature maps. It also takes fully connected CRF [18] to interact with pixels to accomplish lane edge localization and classification precisely. But its huge computation is boring for real-time applications. For better efficiency, the authors in [17] propose spatial CNN (SCNN), which limits communication from slice to slice and not pixel to pixel. Every layer takes former input to apply convolution operation and nonlinear activation, and sends result to the next layer sequentially. Similarly, SCNN treats rows or columns of feature maps to communicate with each other. So, it reduces computation greatly compared with that in [39]. But its computation speed is lower than 10 frames per second.

(2) Row-wise or column-wise segmentation based on the anchor. Lane detection based on pixel-wise segmentation [40–42] requires more computational cost, and it also cannot cope with challenging conditions such as severe occlusion and extreme lighting conditions because of its limited receptive field. For that reason, the authors in [43] propose a row-wise DNN network oriented on row anchors. Its backbone is based on ResNet. Lane detection is described as selecting certain cells. Its loss functions include classification loss, location loss, and structure loss. The row anchors are predefined and include $w + 1$ dimensions. So it can pay more attention to global information and contextual information. The computation cost is closely connected with anchor numbers, anchor dimensions, and lane quantity; it has nothing to do with image pixels. Therefore, it reduces computation cost greatly and promotes lane detection accuracy in no-visual-clue condition [44]. In some studies, the authors put forward a sparse top-down formulation with a large receptive field opposite a down-top formulation in the segmentation network [28, 45–47]. The reason is that traditional segmentation networks have some

shortcomings, such as its computation speed is much slower and has a no-visual-clue problem. To resolve it, a hybrid anchor framework including row anchor-driven and column-anchor-driven representations are proposed, where the former is better for ego lane detection and the latter is right for side lane detection. To cope with global information, it proposes ordinal classification losses, including base classification loss and mathematical expectation loss. The space between classes is continuous.

2.3. Lane Detection Based on Attention Mechanism. The authors in reference [48] propose an attention-guided lane detention model. It utilizes different backbones to extract features such as ResNet-18, ResNet-34, ResNet-101, and so on. But extracting feature maps by means of a DCNN network like the ResNet model easily results in a narrow receptive field. So it adopts a self-attention mechanism to produce a weight vector for every local feature vector. Finally, it implemented matrix multiplication to obtain a global feature map. By doing this, it can predict the lane's existence and its position under conditions of occlusion. [49] proposes expanded-self attention (ESA) module to extract global contextual information. Its main purpose is to divide ESA into HESA (horizontal expanded-self attention) and VESA (vertical expanded-self attention), respectively. Every one predicts the probability of lanes along the horizontal and vertical directions. It is easily seen that it enlarges the receptive field and acquires global contextual information. So it can promote lane detection accuracy, especially in occlusion scenarios.

3. Proposed Approach

In this section, we put forward a lane detection network based on multihead self-attention. Meanwhile, it combines a typical DCNN network such as ResNet-34 with two prediction subnetworks, one for classification and another for regression.

3.1. System Overview. Lane lines represent all sorts of different shapes, types, and colors. For example, it includes solid lines, dotted lines, straight lines, curve lines with different curvatures, emerging lines, and splitting lines. Besides those, some challenging conditions are difficult to handle, such as heavy shadow, severe mark degradation, and vehicle occlusion. Although DCNN is capable of extracting feature maps with convolutions and pooling operations with different kernel sizes and strides, but pooling operations enlarge the receptive field while causing large position offsets. So it requires a trade-off between receptive field, classification, and position accuracy, especially for challenging conditions.

For that reason, we design a multihead self-attention mechanism which taking feature maps of DCNN as inputs. In order to obtain global information, we utilize multiheader to match anchor vectors in different spatial positions. Every head represents global contextual and semantic information among anchors, as shown in Figure 2. So it can summarize and fuse all the global information equally to expand receptive field. Therefore, it also improves classification and location accuracy after sending global anchors to prediction networks.

3.2. Network Design

(1) Backbone: its backbone is ResNet-34, imported from torchvision.models.ResNet-34 which has four layers and one fully connected layer. Each layer has different residuals, which are three, four, six, and three, respectively. Its convolution kernel is three multi three. The channel numbers are 64, 128, 256, and 512 separately. The output of ResNet-34 is a feature map $\mathbf{a}'_{\text{local}}\in \mathbf{R}^{C_b\times H_b\times W_b}$. For reducing dimension and computation cost, it applies 1×1 convolution onto it and generates channel-wise feature map $\mathbf{a}_{\text{local}} \in \mathbf{R}^{C'_b \times H_b\bullet W_b}$.

(2) Multiheader self-attention: we propose $\mathbf{a}_{\text{local}} \in \mathbf{R}^{N\times H_b\bullet W_b}$, $\mathbf{a}_{\text{local}} = [\mathbf{a}_0^{\text{local}}, \mathbf{a}_1^{\text{local}}, \ldots, \mathbf{a}_i^{\text{local}}, \ldots, \mathbf{a}_k^{\text{local}}, \ldots, \mathbf{a}_{N-1}^{\text{local}}]^{\mathrm{T}}$ and N is the number of anchors. The points of the feature map $\mathrm{a}_{\text{local}}$ are composed of anchors. Every row anchor is represented by (x_i, y_i) coordinate frame where $y_i\,(i = 1, 2, \ldots, N_n)$ is equally spaced and predefined. $x_i\,(i = 1, 2, N_n)$ is offset, which is the horizontal distance between the prediction line and the anchor line. N_n is the predefined number in Y direction. It is easily seen that a multihead mechanism can project the d dimension queries, keys, and values h times including different and learned linear projection matrices to get d' dimension queries, keys, and values, such as

$$\mathrm{head_i} = \text{Self Attention}\left(\mathbf{a}_{\text{local}}\bullet\mathbf{W}_{\mathbf{i}}^{\mathbf{Q}}, \mathbf{a}_{\text{local}}\bullet\mathbf{W}_{\mathbf{i}}^{\mathbf{K}}, \mathbf{a}_{\text{local}}\bullet\mathbf{W}_{\mathbf{i}}^{\mathbf{V}}\right). \tag{1}$$

In self-attention mechanism, we compute it by modified dot-product attention, which scales the dot products by $1/\sqrt{d_k}$. d_k is represented by $W_b\bullet H_b$. The process likes as follows:

$$\text{Self Attention}\,(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{soft max}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V}. \tag{2}$$

After we perform the attention function in parallel, they will be concatenated as follows:

$$\text{MultiHead}\,(\mathbf{Q}_{\text{multihead}}, \mathbf{K}_{\text{multihead}}, \mathbf{V}_{\text{multihead}}) = \text{Concat}\,(\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_\mathrm{n}). \tag{3}$$

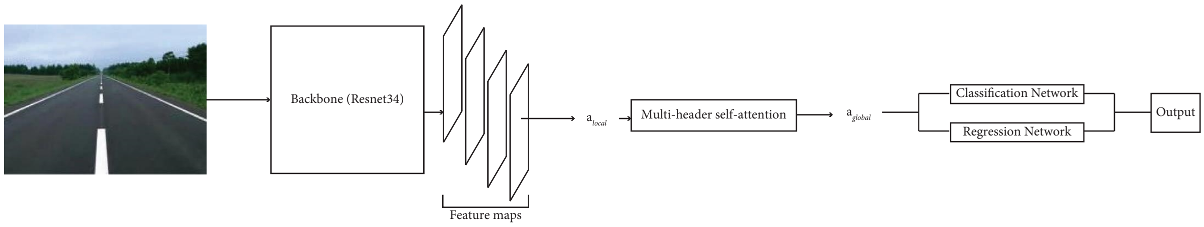

Figure 2: Architecture of the proposed network.

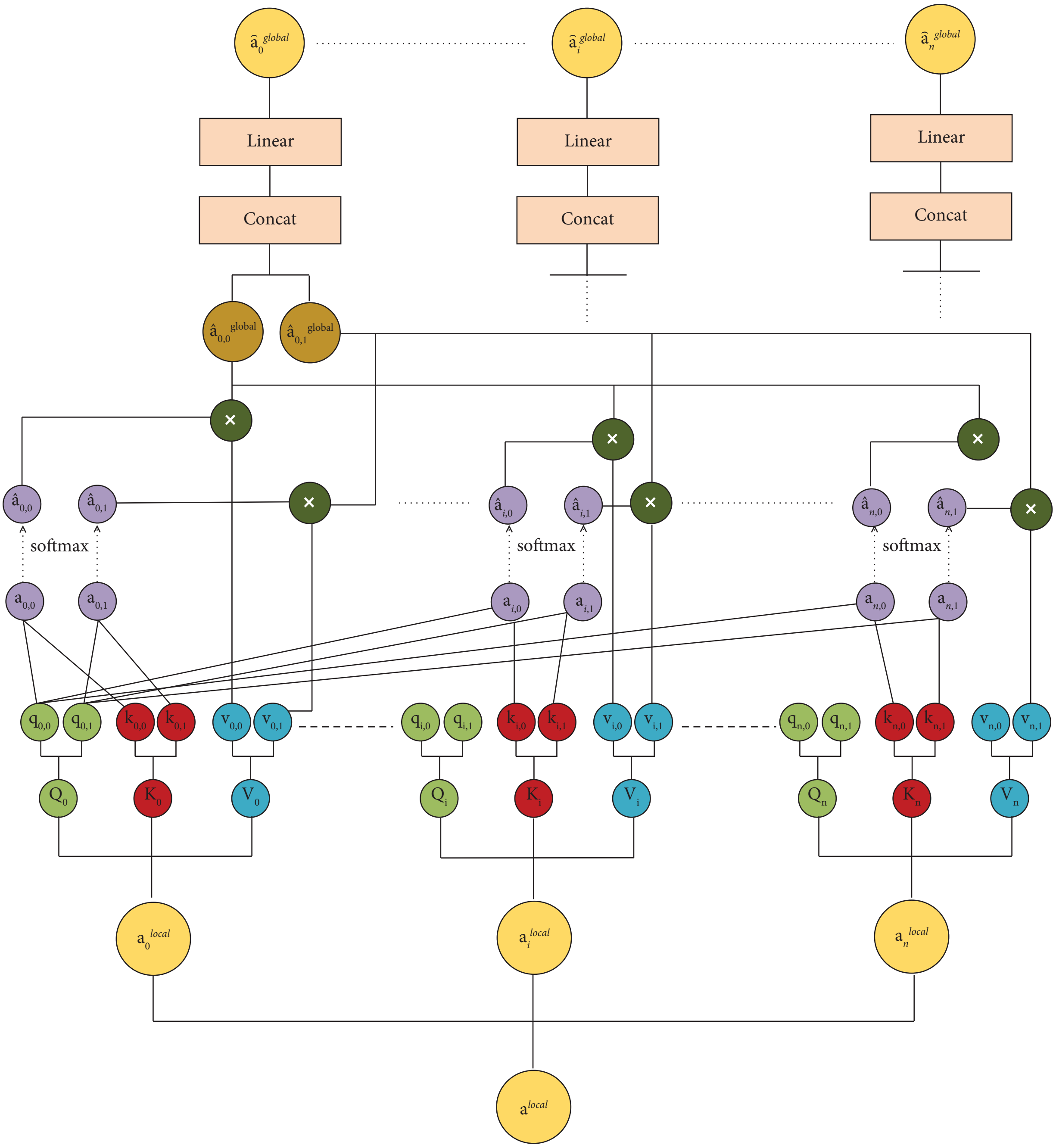

Figure 3: Architecture of multihead self-attention (header is 2).

Finally, we will apply linear projection on multihead to get $\mathbf{a}_{\text{global}}$ as follows:

$$\mathbf{a}_{\text{global}} = \text{MultiHead}\left(\mathbf{Q}_{\text{multihead}}, \mathbf{K}_{\text{multihead}}, \mathbf{V}_{\text{multihead}}\right)\bullet\mathbf{W}^{c}, \tag{4}$$

where the projections are matrices as follows: $\mathbf{W}_i^Q \in \mathbf{R}^{H_b\bullet W_b\times H_b\bullet W_b}$, $\mathbf{W}_i^K \in \mathbf{R}^{H_b\bullet W_b\times H_b\bullet W_b}$, $\mathbf{W}_i^V \in \mathbf{R}^{H_b\bullet W_b\times H_b\bullet W_b}$, and $\mathbf{W}^c \in \mathbf{R}^{8\bullet H_b\bullet W_b\times H_b\bullet W_b}$. Every $\mathbf{head}_i \in \mathbf{R}^{N\times H_b\bullet W_b}\,(i = 1, 2, n)$ and n is the number of heads shown as Figure 3. We also notice that $\mathbf{MultiHead} \in \mathbf{R}^{N\times 8\bullet W_b\bullet H_b}$. So, $\mathbf{a}_{\text{global}}$ has the same dimension as $\mathbf{a}_{\text{local}}$.

(3) Classification model and regression model: before coming into the classification and regression models, we will concatenate $\mathbf{a}_{\text{local}}$ and $\mathbf{a}_{\text{global}}$ together. Also, it becomes an augmented feature vector $\mathbf{a}_{\text{aug}} \in \mathbf{R}^{2\times H_b\bullet W_b}$. So, it will be pushed into the classification model $\mathrm{L}_{\text{class}}$ and regression model $\mathrm{L}_{\text{regression}}$. Finally, $\mathrm{L}_{\text{class}}$ predicts lane line probability $\mathbf{C}_i = \left\{c_0, c_1, \ldots, c_i, \ldots, c_K\right\}$. There are $K + 1$ probabilities all together while K represents the number of lane line and another class is for background or invalid proposal. $\mathrm{L}_{\text{regression}}$ predicts the offset set $\mathbf{l}_i = \left(r, \left\{x_0, x_1, \ldots, x_i, \ldots, x_{N_n-1}\right\}\right)$. r is the number of valid offsets in x direction.

(4) Loss function: in the process of training, we find that the easy negatives can overwhelm training and lead to degenerate models. To resolve it, we propose focal_loss [49, 50] to act as the loss function of the classification model and it follows as this:

$$\begin{aligned}\mathrm{L}_{\text{class}}\left(p_t\right) &= \text{focal\_loss}\left(p_t\right),\\ &= -\alpha_t\left(1 - p_t\right)^{\gamma}\log p_t.\end{aligned} \tag{5}$$

In our paper, we set $\alpha_t = 0.25$ and $\gamma = 2$. For regression model, we adopt Smooth L1 as the loss function. So, our loss function for training combines those two loss functions together. It is defined as follows:

$$\text{Loss}_{\text{total}}\left(\mathbf{c}_i, \mathbf{l}_i\right) = \omega\bullet\sum_{i=0}^{N_{n-1}}\mathrm{L}_{\text{class}}\left(\mathbf{c}_i, \mathbf{c}_i^*\right) + \sum_{i=0}^{N_{n-1}} L_{\text{regression}}\left(\mathbf{l}_i, \mathbf{l}_i^*\right), \tag{6}$$

$\mathbf{c}_i, \mathbf{l}_i$ represent prediction output of classification and regression for anchor i, respectively, and $\mathbf{c}_i^*, \mathbf{l}_i^*$ are ground truth of anchor i. For balancing factor ω, we set $\omega = 10$.

4. Experiments

The widely-used TuSimple [51] and CULane lane detection datasets are used to evaluate our model. In the TuSimple dataset, there are 6,408 annotated images. We split it into a training set (3,268), a validation set (358), and a test set (2,782). The maximum lane marking number is 5. In the CULane [29, 52] dataset, it is also split into a training set (88,880), a validation set (9,675), and a test set (34,680). The maximum lane marking number is 4.

4.1. Implementation Details. Every input image resolution is $H_b \times W_b = 360 \times 640$. It takes 15 epochs on CULane and 100 epochs on TuSimple, whose number of images is less than the former. The learning rate is set at 0.0003, the batch size is set at 8, the total anchor number N is set as 1000, and the offset number N_n is set at 72. All experiments are computed on a personal computer with an 11^{th} Gen Inter(R) Core(TM) i7-11700@2.5 GHz and NVIDIA GeForce GTX 1660 SUPER.

4.2. TuSimple Dataset

4.2.1. Dataset Introduction. The TuSimple dataset includes 6,408 clips, where every clip consists of 20 frames collected in one second. The last frame is labeled with lane ground truth. All the images are of forehead driving scenarios on the high way. The annotations and testing are focused on the current and left/right lanes.

4.2.2. Evaluation and Testing Metrics. In order to compare the performance with other methods, we calculate the accuracy using default TuSimple metrics. It is as follows:

$$\text{Accuracy} = \frac{\sum_{\text{clip}} P_{\text{clip}}}{\sum_{\text{clip}} T_{\text{clip}}}, \tag{7}$$

where P_{clip} is the number of true prediction lane points in current clip and T_{clip} is the total number of ground truth lane points. A lane point is taken as a true positive if its distance from the corresponding label lane point is less than or equal to 15 pixels. While those lane points with distance greater than 20 are taken as negatives. Between them false positives and false negatives are reported and anchors are also dropped. The testing results of the multihead lane detection model based on the TuSimple dataset are shown at Figure 4.

4.2.3. Results. To verify the accuracy of our model, we compare it with several state-of-the-art models. We choose different backbones, such as ResNet-18 and ResNet-34. The qualitative results are shown in Table 1. We know that lane marker detection is extensively applied in real-time conditions. So, it needs high requirements for real-time. From Table 1, we can easily see that the runtime speed of our proposed model can reach from 167.5 to 170.2. Generally speaking, the camera frame rate is about 30 to 60 or so. So it can cope with it, and it will not cause the jam. More importantly, the algorithmic flow of auto-driving consists of perception, prediction, planning, and controlling. From perception to planning, generally, it cannot surpass 100 ms. Therefore, it is better not to exceed 25 ms. The FPS of our proposed models is between 5.875 ms and 5.970 ms. It is only 23.5% to 23.88%. Consequently, it can satisfy the real-time requirements. But because the scenarios in the TuSimple dataset are not relatively complex, our proposal model has a huge amount of room to improve.

4.3. CULane Dataset

4.3.1. Dataset Introduction. The CULane dataset [52] comprises 55 hours of videos consisting of urban, highway, and rural

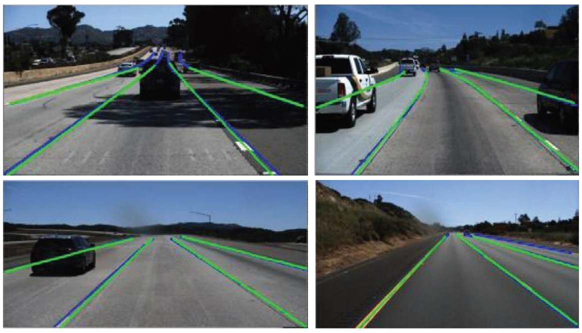

Figure 4: Examples of generated scenes from a multihead lane detection model based on the TuSimple dataset.

Table 1: Comparison among different lane mark detection models based on the TuSimple dataset.

Model	Accuracy (%)	FP	FN	FPS
ResNet-18 [31]	92.69	0.0948	0.0822	**312**
ResNet-34 [31]	92.84	0.0918	0.0796	169
ENet [32]	93.02	0.0886	0.0734	135.4
2-head self-attention (ours)	**95.76**	0.0407	**0.0301**	170.2
4-head self-attention (ours)	95.55	**0.0339**	0.0329	169.5
8-head self-attention (ours)	95.49	0.0414	0.0311	167.5

The significance of bold values means they are the most accurate or they are the lowest error rate.

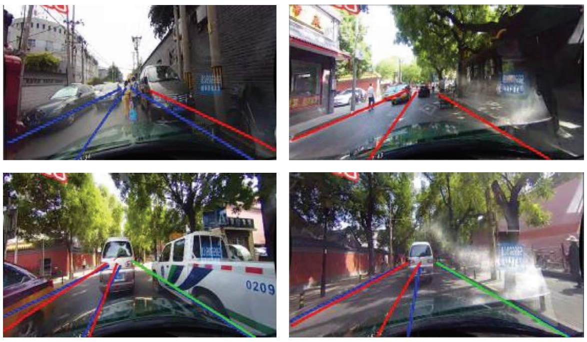

Figure 5: Examples of generated scenes from a multihead lane detection model based on the CULane dataset.

scenarios. All the images have a resolution of 1640 × 590. There are 133,235 frames in total. They are split into a training set that has 88,880 frames, 9,675 for validation, and 34,680 for testing. The test set includes 9 challenging driving scenarios, such as normal, crowd, highlight, shadow, arrow, curve, cross, night, and no line.

4.3.2. Evaluation and Testing Metrics. For judging whether a model detects a lane marker correctly, the metric is the F1 according to the CULane dataset's official references. It considers lane marking as a line with 30 pixel width. So, predictions whose IoUs are greater than 0.5 are treated as true positives. The testing results of multihead lane detection model based on CULane dataset are shown as Figure 5. The metric F_1 – measure is given as follows:

$$
\begin{aligned}
F_1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \\
\text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}.
\end{aligned} \tag{8}
$$

5. Results

The results of our model, along with those of other state-of-the-art models, are shown in Table 2. We know that CULane dataset is much complex compared with TuSimple dataset. It has more challenging scenarios, such as crowd, highlight, shadow, and night. So, we also see that our proposal model is best in challenging scenarios, such as crowds, highlights, and nights. In most challenging scenarios, we achieve better results, except in the shadow scenario alone. We also know that lane marker detection is sensitive to time. From the results of CULane dataset, we can see that FPS of our proposed models is between 167.8 and 170.5 about. That is to say that it takes 5.865 milliseconds to 5.959 milliseconds from getting image input to outputting lane marker points. From the previous analysis, we can also easily see that it can satisfy not only the camera frame rate but also the real-time requirements in auto-driving scenarios.

5.1. Ablation Study. This experiment evaluates the impact of the different-head self-attention mechanism in our proposed model. In Table 3, we can easily see that the 2-head self-attention model achieves the highest accuracy, which is 95.76%. But every different-head self-attention model shows no obvious difference in accuracy. It grows up only 0.33% between highest one and lowest one. In Table 4, we also see that 8-head self-attention overwhelms all other proposal models while increasing 0.12% on F1. The 2-head self-attention model achieves the highest recall while leading 0.12% compared with the other two proposal models. On precision 8-head self-attention, it outperforms other proposal models, rising by about 0.47%. Analyzing the results from the TuSimple dataset, we chose 8-head self-attention to act as our lane detection model. Our main purpose is F1 and precision. But we also know the difference is not obvious.

Table 2: Comparison among different lane mark detection models based on the CULane dataset.

Model	Total	Normal	Crowd	Highlight	Shadow	Arrow	Curve	Cross	Night	No line	FPS
SCNN [17]	71.60	90.60	69.70	58.50	66.90	84.10	64.40	1990	66.10	43.40	7.5
ERF-Net [29]	73.10	**91.50**	71.60	66.01	**71.30**	**87.20**	**71.60**	2199	67.10	45.10	85.87
R-34-SAD [28]	70.70	89.90	68.50	59.90	67.70	83.80	66.02	1960	64.60	42.20	75
R-34-E2E [30]	71.50	90.40	69.90	61.50	68.10	83.70	69.80	2077	63.20	45.01	—
2-head self-attention (ours)	75.43	91.46	73.62	**66.24**	64.07	87.09	66.19	1329	69.96	**48.68**	**170.5**
4-head self-attention (ours)	75.52	91.34	73.56	66.18	66.81	86.79	65.60	**1115**	**70.30**	47.89	169.6
8-head self-attention (ours)	**75.55**	91.43	**73.85**	66.19	69.68	87.02	65.81	1286	69.85	48.18	167.8

The significance of bold values means that F1 is the most highest one.

Table 3: Ablation study results on the TuSimple dataset.

Model	Accuracy (%)	FP	FN
2-head self-attention	95.76	0.0407	0.0301
4-head self-attention	95.55	0.0339	0.0329
6-head self-attention	95.43	0.0350	0.0335
8-head self-attention	95.49	0.0414	0.0311

Table 4: Ablation study results on the CULane dataset.

Model	TP	FP	FN	Precision (%)	Recall (%)	F1 (%)
2-head self-attention	72766	15281	32120	82.64	69.37	75.43
4-head self-attention	72636	14839	32250	83.03	69.25	75.52
8-head self-attention	72649	14762	32237	83.11	69.26	75.55

6. Conclusion

In this paper, we propose a lane marker detection network based on multihead self-attention. It combines row anchoring with multihead self-attention to extract global information to resolve challenging scenarios like vehicle occlusion. It also achieves state-of-the-art performance. On the TuSimple dataset, our proposal method achieves the second-highest accuracy while being much faster than the top-F1 method [28]. On the CULane dataset, our proposal method outperforms other methods. In addition to this, we also find that our proposed approach can be used widely in image classification problems. In [53], it segments the pap smear image using the appropriate threshold. A texture descriptor is proposed titled modified uniform local ternary patterns (MULTP). Then, an optimized multilayer feed-forward neural network is used to classify the pap smear images. The proposed deep neural network is optimized using a genetic algorithm in terms of the number of hidden layers and hidden nodes. In [54], a new version of local binary pattern, that is called completed local quartet patterns, is proposed to extract fabric image local texture features [53, 54] have enough relation. Although we put forward a proposed lane mark detection model, there are some limitations. For example, how to make every head independent in order to focus different subspace and how to set rational anchor number and offset number all need further research. Besides that, it also needs to trade off computation efficiency and computation complexity in the model. For a much better way in the future, we will search for a new architecture synthetically combining encoder-decoders, RNNs, and GANs.

Data Availability

Previously reported TuSimple and CULane data were used to support the findings of this study and are available at https://github.com/TuSimple/tusimple-benchmark and https://xingangpan.github.io/projects/CULane.html. These prior studies and datasets are cited at relevant places within the text as references [16, 41, 47–52].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] Z. Qin, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Transactions on Vehicular Technology*, vol. 41-54, p. 1, 2020.

[2] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deepcrack:Learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, pp. 1–15, 2018.

[3] L. Chen, Q. Li, Q. Mao, and Q. Zou, "Block-constraint line scanning method for lane detection," in *Proceedings of the IEEE Intell. Vehicles Symp*, pp. 89–94, Aachen, Germany, July 2010.

[4] J. M. Trivedi, "Video-based lane estimation and tracking for driver Assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[5] A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Proceedings of the International Conference on Computer and*

Communication Engineering, Kuala Lumpur, Malaysia, September 2008.

[6] R. F. Berriel, E. de Aguiar, F. Alberto, D. Souza, and T. Oliveira-Santos, "Ego-lane analysis system (ELAS): dataset and algorithms," *Image and Vision Computing*, vol. 68, no. 64–75, p. 1, 2017.

[7] Z. Wang, W. Ren, and Q. Qiu, "LaneNet: real-time lane detection networks for autonomous driving," 2018, https://arxiv.org/abs/1807.01726.

[8] L. Chen, Q. Li, Q. Mao, and Q. Zou, "Block-constraint line scanning method for lane detection," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 89–94, Aachen, Germany, July 2010.

[9] Q. Zou, L. Ni, Q. Wang, Q. Li, and S. Wang, "Robust gait recognition by integrating inertial and rgbd sensors," *IEEE Transactions on Cybernetics*, vol. 48, no. 4, pp. 1136–1150, 2018.

[10] Y.-C. Hsu, Z. Xu, Z. Kira, and J. Huang, *Learning to Cluster for Proposal-free Instance Segmentation*, pp. 1–8, IJCNN, Padua, Italy, 2018.

[11] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," 2019, https://arxiv.org/abs/1902.00293#:%7E:text=End%2Dto%2Dend%20Lane%20Detection%20through%20Differentiable%20Least%2DSquares%20Fitting,-Wouter%20Van%20Gansbeke%26text=Lane%20detection%20is%20typically%20tackled,the%20post%2Dprocessed%20mask%20next.

[12] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Polylanenet: lane estimation via deep polynomial regression," in *Proceedings of the Int. Conf. Pattern Recog*, pp. 6150–6156, IEEE, Milan, Italy, July 2021.

[13] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, *End-to-end Lane Shape Prediction with Transformers*, pp. 3694–3702, WACV, Waikoloa, HI, USA, 2021.

[14] V. Badrinarayanan, A. Kendall, and R. C. Segnet, "A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, p. 1, 2017.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention*, Strasbourg, France, July 2015.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[17] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: spatial CNN for traffic scene understanding," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial*, vol. 02, pp. 7276–7283, Washington, DC. USA, June 2018.

[18] P. Lu, C. Cui, S. Xu, H. Peng, and F. Wang, "SUPER: a novel lane detection system," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 583–593, 2021.

[19] W. Wang, H. Lin, and J. Wang, "CNN based lane detection with instance segmentation in edge-cloud computing," *Journal of Cloud Computing*, vol. 3, 2020.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2014.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," 2015, https://arxiv.org/abs/1505.04597.

[22] Y. Guo, G. Chen, P. Zhao et al., "Gen-LaneNet: a generalized and scalable approach for 3D lane detection," *Computer Vision and Pattern Recognition*, vol. 3, 2020.

[23] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3D-lanenet: end-to-end 3D multiple lane detection," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, Montreal, QC, Canada, June 2019.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[25] M. Bai, G. Mattyus, N. Homayounfar, S. Wang, and R. Urtasun, "Deep multi-Sensor lane detection," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, June 2018.

[26] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*CVPR, New Orleans, LA, USA, 2016.

[27] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Adv. Neural Inform. Process. Syst*, pp. 5998–6008, Cambridge, Massachusetts, June 2017.

[28] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNs by self attention distillation," in *Proceedings of the Int. Conf. Computer.Vis*, pp. 1013–1021, Cambridge, MA, USA, June 2019.

[29] L. Tong, Z. Chen, Y. Yang, Z. Wu, and H. Li, "Lane Detection in low-light conditions using an efficient data enhancement: Light conditions style transfer," vol. 1, no. 2, pp. 6–8, 2020, https://arxiv.org/abs/2002.01177.

[30] S. Yoo, H. Lee, M. Heesoo et al., "End-to-End lane marker detection via row-wise classification," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 07, New Orleans, LA, USA, June 2020.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, San Juan, PR, USA, June 2016.

[32] A. Paszke, A. Chaurasia, S. Kim, and E. C. Enet, "A deep neural network architecture for real-time semantic segmentation," vol. 2, no. 7, p. 10, 2016, https://arxiv.org/abs/1606.02147.

[33] A. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Mach. Vision Appl*, vol. 25, no. 3, pp. 727–745, 2014.

[34] S. Yenikaya, G. Yenikaya, and E. Dven, "Keeping the vehicle on the road: a survey on on-road lane detection systems," *ACM Computing Surveys*, vol. 46, no. 1, 2013.

[35] M. Aly, "Real time detection of lane markers in urban streets," in *Proceedings of the IEEE Intell. Vehicles Symp*, pp. 7–12, Aachen, Germany, July 2008.

[36] Y. Wang and K. D. Shen, "Lane detection and tracking using b-snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.

[37] S. Zhou, Y. Jiang, J. Xi, J. Gong, G. Xiong, and H. Chen, "A novel lane detection based on geometrical model and gabor filter," in *Proceedings of the IEEE Intell. Vehicles Symp*, pp. 59–64, Aachen, Germany, July 2010.

[38] M. A. Selver, E. Er, B. Belenlioglu, and Y. Soyaslan, "Camera based driver support system for rail extraction using 2-D gabor wavelet decompositions and morphological analysis,"

in *Proceedings of the IEEE Conf. Intell. Rail Transp*, pp. 270–275, Singapore, July 2016.

[39] L.-C. Chen, P. George, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional Nets, atrous convolution, and fully connected CRFs," 2016, https://arxiv.org/abs/1606.00915.

[40] B. Huval, T. Wang, S. Tandon et al., "An empirical evaluation of deep learning on highway driving," 2015, https://arxiv.org/abs/1504.01716.

[41] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards endto-end lane detection: an instance segmentation approach," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 286–291, Aachen, Germany, July 2018.

[42] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: spatial cnn for traffic scene understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7276–7283, Washington, DC. USA, September 2018.

[43] Z. Qin, H. Wang, and X. Li, "Ultra Fast structure-aware deep lane detection," *Computer Vision - ECCV 2020 Computer Vision – ECCV*, vol. 11, pp. 276–291, 2020.

[44] Z. Qin, P. Zhang, and Xi Li, "Ultra Fast deep lane detection with hybrid anchor driven ordinal classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1-14, p. 6, 2022.

[45] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, *Spatial as deep: Spatial Cnn for Traffic Scene Understanding*, pp. 7276–7284, AAAI, Washington, DC. USA, 2018.

[46] S. Lee, J. Kim, J. Shin Yoon et al., "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," in *Proceedings of the Int. Conf. Comput. Vis*, pp. 1947–1955, Montreal, BC, Canada, October 2017.

[47] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: an instance segmentation approach," in *Proceedings of the IEEE Intell. Veh. Symp*, pp. 286–291, Aachen, Germany, July, 2018.

[48] L. Tabelini, R. Berriel, T. M. P. ao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: real-time attention-guided lane detection," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog*, pp. 1–9, Seattle, WA, USA, June, 2021.

[49] M. Lee, J. Lee, D. Lee, W. Kim, S. Hwang, and S. Lee, "Robust lane detection via expanded self attention," in *Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1949–1958, Waikoloa, HI, USA, June, 2022.

[50] T.-Y Lin, P. Goyal, R. Girsshcik, K. He, and P. Dollar, "Focal loss for Dense Object detection," in *Proceedings of the Conference Computer Vision and Pattern Recognition(CVPR)*, Waikoloa, HI, USA, June 2017.

[51] TuSimple, "Tusimple benchmark," 2022, https://github.com/TuSimple/tusimple-benchmark.

[52] C. U. L. Culane, "Benchmark," vol. 7, no. 12, p. 13, 2022, https://xingangpan.github.io/projects/CULane.html. Accessed.

[53] S. Fekri-Ershad 1 and S. Ramakrishnan, "Cervical cancer diagnosis based on modified uniform local ternary patterns and feed forward multilayer network optimized by genetic algorithm," *Computers in Biology and Medicine*, vol. 144, p. 5, 2022.

[54] Z. Pourkaramdel, S. Fekri-Ershad, and L. Nannic, "Fabric defect detection based on completed local quartet patterns and majority decision algorithm," *Expert Systems with Applications*, vol. 198, p. 7, 2022.