Research Article

A Network That Balances Accuracy and Efficiency for Lane Detection

Ce Zhang , Yu Han, Dan Wang, Wei Qiao, and Yier Lin

College of Mechanical Engineering, Tianjin University of Science & Technology, Tianjin, China

Correspondence should be addressed to Yier Lin; yier.lin@tust.edu.cn

Received 24 September 2021; Revised 15 November 2021; Accepted 20 November 2021; Published 7 December 2021

Academic Editor: Fazlullah Khan

Copyright © 2021 Ce Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the automatic lane-keeping system (ALKS), the vehicle must stably and accurately detect the boundary of its current lane for precise positioning. At present, the detection accuracy of the lane algorithm based on deep learning has a greater leap than that of the traditional algorithm, and it can achieve better recognition results for corners and occlusion situations. However, mainstream algorithms are difficult to balance between accuracy and efficiency. In response to this situation, we propose a single-step method that directly outputs lane shape model parameters. This method uses MobileNet v2 and spatial CNN (SCNN) to construct a network to quickly extract lane features and learn global context information. Then, through depth polynomial regression, a polynomial representing each lane mark in the image is output. Finally, the proposed method was verified in the TuSimple dataset. Compared with existing algorithms, it achieves a balance between accuracy and efficiency. Experiments show that the recognition accuracy and detection speed of our method in the same environment have reached the level of mainstream algorithms, and an effective balance has been achieved between the two.

1. Introduction

In ALKS, the vehicle must reliably detect the boundary of its current lane for precise positioning, and on this basis, the understanding of the traffic scene is completed, and the vehicle is kept in the lane through trajectory planning and vehicle control. Among them, the lane detection module is the starting point of the entire system, and its safety and effectiveness are particularly important. However, due to the inherent slenderness and complex conditions of the lane itself, such as weather changes, light changes, and other road users' occlusion, this task becomes very challenging. In addition, the calculation time of the detection module is very important for the real-time application of the entire system. It is necessary to improve the high operating efficiency and transmission adaptability of the algorithm while maintaining detection accuracy [1].

The early algorithms are mostly traditional methods [2, 3], which mainly used the fusion of manual features and heuristic methods, and then combined postprocessing techniques, such as Hough transform (detecting lines,

circles, or other parametric curves), random sample consensus (RANSAC) estimating parameters of a mathematical model from a set of observed data that contains outliers, and other algorithms for lane line detection. This type of algorithm has a small amount of calculation but requires manual adjustment of parameters, a large workload, and poor robustness. When the driving environment changes significantly, the detection effect of lane lines is not good. The current deep learning-based methods have become the current mainstream method due to their high accuracy. Among them, the method based on instance segmentation [4] is to first generate the segmentation results, and then perspective transformation is used to convert into a bird's-eye view (BEV) and perform curve-fitting. Popular models include polynomials, splines, or cyclotrons. This type of method achieves high accuracy by automatically extracting features from data, but the low-efficiency decoder makes the calculation efficiency low and is not sensitive enough to the curve scene [5, 6]. In response to this situation, the method based on message transmission [7, 8] uses spatial information in deep neural networks to capture the global context

to improve recognition accuracy. However, a major problem is still that this method is usually computationally intensive and difficult to run in real-time, which is not conducive to the use of ALKS in-vehicle embedded devices. In addition, there is a method based on end-to-end training [9], which directly returns lane line parameters. This method has a great improvement in speed, but it is slightly insufficient in accuracy and lacks overall interpretability.

To effectively balance the accuracy and efficiency of the algorithm, we propose a single-step lane detection method based on the MobileNet v2 + SCNN network. The network backbone of the model adopts the lightweight MobileNet v2 [10], which can effectively reduce the calculation amount and parameters of the lane model. At the same time, the SCNN layer is added so that the information can be effectively transmitted in the space layer. The network finally outputs the polynomial coefficients of each lane and the confidence score of each lane. Through experiments, the recognition accuracy and detection speed of the model proposed in this paper have reached the level of mainstream algorithms in the same environment, and an effective balance has been achieved between the two.

2. Related Work

Lane detection methods can be based on traditional image processing and deep learning. We briefly review the most extensive classical methods related to lane detection and highlight the differences between them in this section.

2.1. Traditional Methods. Traditional lane detection methods usually use road image features, geometric features, and another information modeling [11]. Feature extraction is based on global and local location information [12, 13]. Feature extraction is essentially a filtering algorithm which aims to reduce the number of features in a dataset by creating new features from the existing ones. Then, Hough transform and RANSAC are used for straight line and curve-fitting, respectively. Finally, the error detection is eliminated, and the lane boundary segments are clustered into the final result. Traditional lane line detection methods cannot achieve the function of lane tracking, and most of the algorithms are limited to specific application environments and are not robust to light changes and sudden weather changes. Similarly, there are lane color degradation and structural damage in real data, as well as road noise and occlusion. Traditional methods cannot handle the complex situations faced in actual driving.

2.2. Deep Learning Methods. Lane detection methods based on deep learning can be roughly divided into two categories: single-step models and two-step models. The two-step model first extracts the characteristics of the lane lines and then clusters and fits each line. The feature extraction in the first stage is mostly based on segmentation. For example, VPGNet [14] uses four-quadrant segmentation to define the location of the vanishing point and guides network learning through the vanishing point, to obtain a better convergence

effect, and a model converges when additional training will not improve the model. SCNN stacks convolutional layers so that information can be transmitted across rows and columns. It is effective for long and narrow lane detection, but the running speed is only 7.5 FPS. The authors proposed a self-attention distillation (SAD) module in [15]. Based on information distillation, to solve the impact of the large backbone network on the speed, text information can be aggregated. CurveLanes-NAS [16] down samples the entire image and classifies (grid) rows of each cell. Although the most advanced results have been achieved, they are computationally very time-consuming. In the second stage of the two-step model, most of the work is performed on curve-fitting through the learned transformation matrix. First, the result of the first segmentation is converted into a BEV, and then uniform point selection + least squares is used to fit the line of the mask map.

The single-step model can directly output the parameters of the lane shape model, for example, Line-CNN [17], LaneATT [18], and so on. LaneATT is implemented based on anchors, and the attention mechanism is applied to perform SOTA, reaching 250 FPS. In addition, PolyLaneNet [9] assumes that the lane line is a curve and uses polynomials to learn curve parameters. However, due to the imbalance of the existing dataset, some deviations are caused.

The rest of the paper is organized according to the following pattern. Methods and materials are discussed in Section 3, Results are given in Section 4, and the paper is concluded in Section 5.

3. Method

We will describe the structure and loss function of our proposed single-step lane detection method based on the MobileNet v2 + SCNN network in this section.

3.1. Architecture Design. The proposed network structure is shown in Figure 1. It mainly consists of two parts including the backbone network and the SCNN layer, and then the fully connected layer outputs the prediction results.

Lightweight MobileNet v2 uses deep separable convolution instead of ordinary convolution, which can reduce the amount of calculation and parameters of the model. In addition, it draws on the residual connection idea of ResNet [19] and proposes a reverse residual structure on this basis. The number of network layers is deepened, and the ability to express features is enhanced; a linear bottleneck structure is used to replace nonlinear bottlenecks to reduce the loss of low-dimensional feature information. Because of these advantages, we use MobileNet v2 as the network backbone, discard the last two fully connected layers, and replace them with a dilated convolutions layer. Dilated convolutions can get a larger receptive field and thus get denser data. At the same time, the spatial characteristics of the image can be well preserved without the loss of image information. After expanding the dilated convolutions layer, an SCNN layer is added to allow the message to propagate down and to the

FIGURE 1: Architecture of the proposed network. The double SCNN modules with the suffixes “SCNN_D” and “SCNN_R,” respectively, indicate that they are transmitted in a downward and rightward direction in space.

right so that each pixel can receive messages from all other pixels to further expand the receptive field.

For each frame of the image to be detected, $M_{\max}$ lane marker candidates (expressed as a polynomial), the lowest point of each lane marker, and the uniform vertical height h of the horizon will be output. All output results can be expressed as follows:

$$f(I; \theta) = \left(\{P_j, s_j, c_j\}_{j=1}^{M_{\max}}, h \right), \quad (1)$$

where P_j is the polynomial of the lane marking candidates, s_j is the vertical offset, and $c_j \in [0, 1]$ is the prediction confidence score, as illustrated in Figure 2 [9].

And P_j can be expressed as follows:

$$P_j(y) = \sum_{k=0}^K a_{k,j} y^k, \quad (2)$$

where K is the parameter that defines the order of the polynomial.

3.2. Loss Function. The loss function is a basic and key element in deep learning. To balance the magnitude of the different parts of the loss function and improve the convergence speed, it is necessary to define the loss function by weight:

$$\text{loss} = W_p \text{loss}_p + W_s \text{loss}_s + W_c \text{loss}_c + W_h \text{loss}_h, \quad (3)$$

where the weighting coefficients W_p , W_s , W_c , and W_h are hyperparameters manually adjusted.

The first part of the loss refers to how well the polynomial model fits. Here, we use the mean square error (MSE) function to calculate the error between the predicted value and the true value. The closer the predicted value x_j^* is to the ground true value x_j , the smaller the MSE is between the two. Loss_p is defined as follows:

$$\text{loss}_p = E(x_j - x_j^*)^2, \quad (4)$$

where $x_j^* = [x_{1,j}^*, \dots, x_{N,j}^*]^T$ and $x_j = [x_{1,j}, \dots, x_{N,j}]$.

Next, the second part of the loss is the vertical offset s_j loss. The last part of the loss is the vertical position h loss. We

also use the MSE to calculate them. The third part of the loss is the prediction confidence score c_j loss. For this binary classification task, we use cross-entropy to calculate it which measures the relative entropy between two probability distributions over the same set of events.

4. Experiments

We will show the verification effect of the proposed single-step lane detection method based on the MobileNet v2 + SCNN network on the TuSimple dataset in this section. Experimental results show that the recognition accuracy and detection speed of the detection algorithm proposed in this paper have reached the level of mainstream algorithms in the same environment, and there is an effective balance between the two. Next, the details of the implementation process will be detailed, and the experimental results will be analyzed.

4.1. Implementation Details

4.1.1. Datasets. In the latest academic papers, most of the TuSimple [20] and CULane datasets are used for performance comparison. Since the related algorithms involved in this article are all tested in TuSimple, this article also uses this dataset for verification. TuSimple is part of the data disclosed in the Autonomous Driving Company TuSimple’s Lane Marking Challenge again. The dataset consists of 72 k 1280×720 images. It collects scenes of high-speed roads with clear weather and clear lane lines. The characteristic is that the lane is marked with points.

4.1.2. Evaluation Metrics. To ensure comparison with other algorithms under the same conditions, we also follow TuSimple’s original evaluation indicators: accuracy (Acc), false negative (FN), and false-positive (FP) rate. The predicted lane marking is true positive, when the distance between the end of the line and its surrounding area is less than 20 pixels.

FIGURE 2: Schematic representation of the lane marking information output by each frame of image.

TABLE 1: List of the results.

Method	ACC (%)	FP	FN	FPS
SCNN	96.53	0.0617	0.0180	7
PolyLaneNet	88.36	0.0942	0.0933	91
SPNet	92.41	0.0775	0.0512	30

FIGURE 3: Visualization on the TuSimple dataset. The green mark is the ground truth, and the blue mark is the prediction.

4.1.3. Hyperparameters. We implemented the proposed network based on Pytorch. The transformations of random cropping, scaling, rotation, and color dithering are used to increase the diversity of images and increase the number of training samples. Then, the image is resized to 1280×720 pixels. Finally, ImageNet's [21] mean and standard deviation are used for normalization. The hyperparameters in the network are set as follows: the batch size is set to 4, the SCNN convolution kernel width is set to 7, the loss coefficients W_S , W_C , and W_h are set to 1, and W_p is set to 800. Cosine annealing is used to adjust the learning rate. The initial learning rate is $3e-4$, and the period is 750 periods. We performed 9.5k iterations on a single Nvidia P4000 GPU and got the final result.

4.2. Results. Based on the same environment of the TuSimple dataset, we verify the effectiveness of our proposed method and compare it with PolyLaneNet and SCNN. In this

experimental result, the accuracy and running time of TuSimple's evaluation are compared.

From Table 1, we can see that the proposed method has achieved the recognition performance of mainstream algorithms. The ACC rate is 1.05% higher than that of PolyLaneNet. Although it does not achieve the same performance as SCNN, it is not far from it. At the same time, it is about 4 times faster than SCNN, which makes it have practical application value in the actual ALKS system. Our method achieves an effective balance between accuracy and efficiency.

In addition to quantitative evaluation, some visualization work has also been done on the TuSimple dataset, and the effect is shown in Figure 3.

5. Conclusion

In this research, we propose a single-step lane detection method based on the MobileNet v2+SCNN network to solve the balance between accuracy and efficiency. The

backbone network is based on the lightweight MobileNet v2, which greatly reduces the amount of calculation and parameters of the lane model. After that, the additional dilated convolutions in the network expand the receptive field, and the SCNN layer realizes the effective transmission of spatial layer information, which ensures the recognition accuracy of the model. The network finally directly outputs the polynomial coefficient of each lane and the confidence score of each lane. Experiments show that the recognition accuracy and detection speed of our method in the same environment have reached the level of mainstream algorithms, and an effective balance has been achieved between the two. In the future, we will deploy this method on mobile devices.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to Lucas et al. for publicly releasing the source CODE (for both training and inference) and the trained models so that their works on lane markings detection have a baseline to start work and for comparison.

References

- [1] J. Philion, "Fastdraw: addressing the long tail of lane detection by adapting a sequential prediction network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11582–11591, Long Beach, CA, USA, June 2019.
- [2] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognition*, vol. 73, pp. 216–234, 2018.
- [3] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust lane detection using two-stage feature extraction with curve fitting," *Pattern Recognition*, vol. 59, pp. 225–233, 2016.
- [4] 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 Intelligent Vehicles Symposium*, pp. 286–291, Changshu, China, June 2018.
- [5] Y. M. Ko, J. Jun, D. Ko, and M. Jeon, "Key points estimation and point instance segmentation approach for lane detection," 2020, <https://arxiv.org/abs/2002.06604>.
- [6] 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*, pp. 7276–7283, New Orleans, LA, USA, February 2018.
- [7] M. Ghafoorian, C. Nugteren, N. Baka, O. Booi, and M. Hofmann, "EL-GAN: embedding loss driven generative adversarial networks for lane detection," in *Proceedings of the European Conference on Computer Vision*, Volume 11129 of Lecture Notes in Computer Science, pp. 256–272, Springer, Munich, Germany, September 2018.
- [8] J. Zhang, Y. Xu, B. Ni, and Z. Duan, "Geometric constrained joint lane segmentation and lane boundary detection," in *Proceedings of the European Conference on Computer Vision*, pp. 502–518, Springer, Munich, Germany, September 2018, Volume 11205 of Lecture Notes in Computer Science.
- [9] L. T. Torres, R. F. Berriel, T. M. Paixao, et al., PolyLaneNet: lane estimation via deep polynomial regression," 2020, <https://arxiv.org/abs/2004.10924>.
- [10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.
- [11] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014.
- [12] Z. Teng, J.-H. Kim, and D.-J. Kang, "Real-time lane detection by using multiple cues," in *Proceedings of the International Conference on Control, Automation and Systems*, pp. 2334–2337, Gyeonggi-do, Republic of Korea, October 2010.
- [13] X. Wang, D. Yan, K. Chen et al., "Lane extraction and quality evaluation: a Hough transform based approach," in *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval*, pp. 7–12, Shenzhen, China, August 2020.
- [14] S. Lee, J.-S. Kim, J. S. Yoon et al., "VPGNet: vanishing point guided network for lane and road marking detection and recognition," 2017, <https://arxiv.org/abs/1710.06288>.
- [15] Y. Hou, M. Zheng, C. Liu, and C. C. Loy, "Learning lightweight lane detection CNNs by self attention distillation," in *Proceedings of the International Conference on Computer Vision*, pp. 1013–1021, Seoul, Republic of Korea, August 2019.
- [16] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "CurveLane-NAS: unifying lane-sensitive architecture search and adaptive point blending," in *Proceedings of the European Conference on Computer Vision*, Glasgow, UK, August 2020.
- [17] Li Xiang, J. Li, X. Hu, and J. Yang, "Line-CNN: end-to-end traffic line detection with line proposal unit," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 248–258, 2020.
- [18] Keep your eyes on the lane: real-time attention-guided lane detection, CVPR, 2021.
- [19] 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, Las Vegas, NV, USA, June 2016.
- [20] TuSimple, "Tusimple dataset," 2017, <https://cvpr2017.tusimple.ai/>.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, USA, June 2009.