Lane Detection Based on Deep Learning

Taojie Wang Mengxuan Wu

Abstract

Lane detection is a crucial task in autonomous driving. It is a challenging task due to the complex road conditions and the diversity of lane markings. Recent advances in deep learning have shown promising results in lane detection. In this project, we reviewed the state-of-the-art deep learning methods for lane detection and propose a new method to improve the performance of lane detection. Additionally, We evaluated the proposed method on various datasets and compared it with existing methods. The goal of this project is to develop an accurate and robust lane detection system for autonomous driving.

1 Introduction

Lane detection is a computer vision task that involves identifying the boundaries of driving lanes on the road. The goal is to accurately detect and track the lane markings in real-time, even under challenging conditions such as poor lighting, occlusions, and varying road surfaces. Lane detection is a crucial component of autonomous driving systems, as it provides information for vehicle control, path planning, and decision-making, which are essential for safe and efficient driving.

Recent advances in deep learning have shown promising results in lane detection. Convolutional neural networks (CNNs) have achieved state-of-the-art performance on various lane detection datasets. These models can learn complex features from raw images and generalize well to unseen road conditions.

In this project, we will review the state-of-the-art deep learning methods for lane detection and propose a new method to improve the performance of lane detection. We plan to evaluate the proposed method on CULane [1], TuSimple [2] and our own dataset collected in SUSTech campus. We will compare the proposed method with existing methods and analyze the results to identify the strengths and weaknesses of different approaches.

2 Related Work

2.1 Lane Detection with CNNs

Recent studies of lane detection with CNNs can be divided into three categories: segmentation-based methods, anchor-based methods, and parameter-based methods. And we will cover them in the following sections.

2.1.1 Segmentation-based Methods

Segmentation-based methods treat lane detection as a semantic segmentation task, where the goal is to classify each pixel in the image to belong to a lane or background. SCNN [3] proposes a message-passing mechanism to address the no visual evidence problem, thereby improving the performance of lane detection. However, it is slow for real-time application. RESA [4] proposes a real-time feature aggregation mechanism to improve the efficiency of lane detection. CurveLane-Nas [5] uses neural architecture search to automatically design a network for lane detection, but it requires a large amount of computational resources. These methods are generally time-consuming, since they need to process each pixel in the image.

2.1.2 Anchor-based Methods

In anchor-based methods, lanes are represented by x-coordinates at each row of the image. LaneATT [6] uses an anchor-based attention mechanism that aggregates global information. UFLD [7] first proposes a row anchor-based lane detection method and adopts lightweight backbones to achieve high inference speed. CondLaneNet [8] introduces a conditional lane detection strategy based on conditional convolution and row anchor-based formulation. CLRNet [9] uses learnable anchor parameters (x, y-coordinates, angle, and width) to represent lanes. CLRerNet [10] develops a novel row detector based on CLRNet and achieves state-of-the-art performance. LaneFormer [11] employs a transformer with row and column attention to detect lane instances in an end-to-end manner. These method are more adopted as they leverages speed and accuracy.

2.1.3 Parameter-based Methods

Parameter-based methods represent lanes as parametric curves or polynomials. PolyLaneNet [12] uses a polynomial regression model to predict lane parameters. LSTR [13] uses transformer-based models to predict the parameter set of lanes. These methods generally have faster inference speed. However, they have difficulty handling complex lane shapes, especially when the lane shape is not well represented by the chosen parametric model.

2.2 Datasets

2.2.1 CULane

CULane [1] is a large scale challenging dataset for academic research on traffic lane detection. It is collected by cameras mounted on six different vehicles driven by different drivers in Beijing. More than 55 hours of videos were collected and 133,235 frames were extracted. The dataset is divided into 88880 for training set, 9675 for validation set, and 34680 for test set. An example of annotated images and categories in CULane dataset is shown in Figure 1.

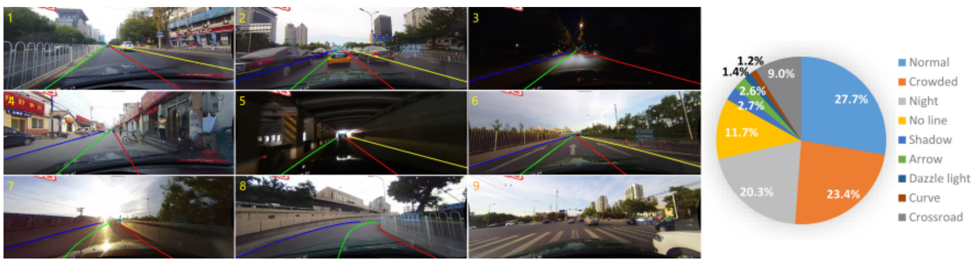

Figure 1: Example annotated images and categories in CULane dataset.

2.2.2 TuSimple

The TuSimple [2] dataset consists of 6,408 road images on US highways. The resolution of image is 1280*720. The dataset is composed of 3,626 for training, 358 for validation, and 2,782 for testing. It is first used in CVPR 2017 Workshop on Autonomous Driving Challenge. An example of annotated images in TuSimple dataset is shown in Figure 2.

Figure 2: Example annotated images in TuSimple dataset.

2.2.3 3D Lane Dataset

In addition to traditional lane detection tasks that requires predicting the lane markings in 2D images, recently, 3D lane detection has attracted increasing attention. In 3D lane detection, the goal is to predict the 3D coordinates of the lane markings in the world coordinate system. The two most popular 3D lane detection datasets are the OpenLane [14] dataset and the Apollo [15] dataset. However, due to the lack of corresponding tools for 3D data collection and annotation, we will focus on 2D lane detection in this project.

3 Method

After reviewing the state-of-the-art deep learning methods for lane detection and collected the CULane and TuSimple datasets, we choose to implement CLRNet, which achieves state-of-the-art performance on the LLAMAS dataset and ranked 4-th on the TuSimple dataset. To further improve the model, we replace the original FPN with the path aggregation feature extraction network (PAFPN) [16].

3.1 CLRNet Model Architecture

In CLRNet, it first performs detection in high semantic features to coarsely localize lanes. Then, it performs refinement based on fine-detail features to get more precise locations. Progressively refining the location of lane and feature extraction leads to high accuracy detection results. To solve the problem of non-visual evidence of lane, it introduces ROIGather to capture more global contextual information by building the relation between the ROI lane feature and the whole feature map. Moreover, it defines the IoU of lane lines and proposes the Line IoU (LIoU) loss to regress the lane as a whole unit and considerably improves the performance compared with standard loss. The architecture of CLRNet is shown in Figure 3.

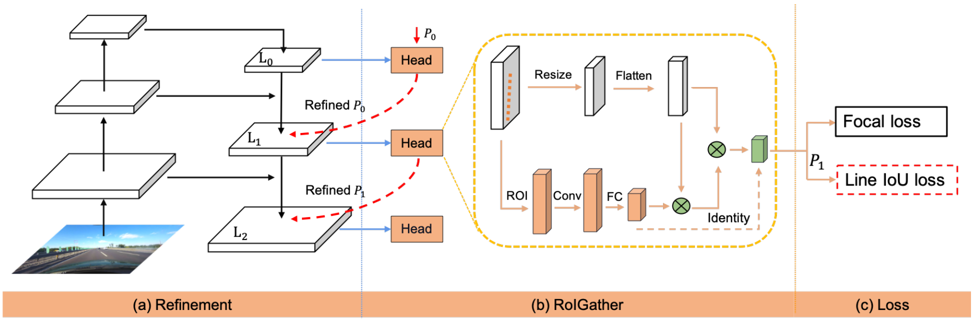

Figure 3: CLRNet architecture.

3.2 PAFPN Backbone

The path aggregation feature extraction network serves as backbone of PANet, which ranked 1st in COCO Instance Segmentation Challenge 2017. Its major modification is adding an extra bottom to top path augmentation network to the original FPN. The motivations for this are that it further augmented the high level features with low level features which helps localize objects, and it also provides high level features a path that consists of less than 10 layers which preserves the features better. The architecture of PANet is shown in Figure 4.

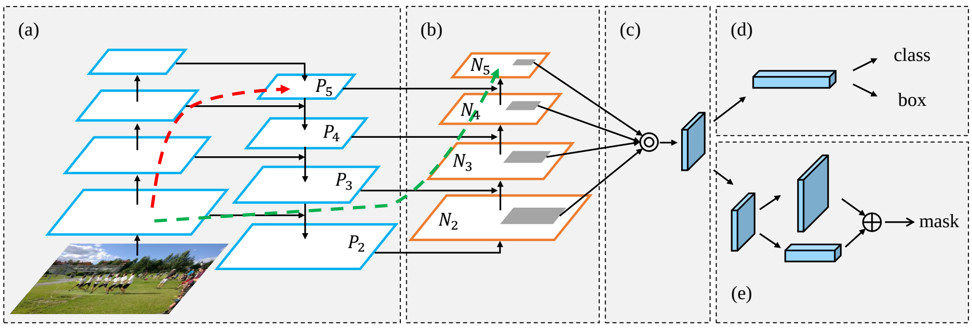

Figure 4: PANet architecture.

3.3 Novelty

In this research, we propose a modification to the feature extraction network of CLRNet by replacing its original Feature Pyramid Network (FPN) with the Path Aggregation Feature Pyramid Network (PAFPN) derived from PANet. We hypothesize that the integration of the path augmentation module introduced by PAFPN will significantly enhance the quality of extracted features, thereby leading to improved overall model performance in terms of accuracy and robustness. Furthermore, we validate the effectiveness of the proposed modifications by conducting experiments on a newly collected dataset of images obtained at SUSTech.

4 Experiment

4.1 Setup

To establish a benchmark for evaluating our proposed approach, we utilize the vanilla CLRNet model with a ResNet-34 backbone as the baseline. Both the baseline model and the modified version incorporating the PAFPN are trained on a single NVIDIA RTX 4090 GPU using the CULane dataset. Detailed descriptions of the dataset, training procedures, and parameter configurations are provided in Appendix A for reproducibility and clarity.

4.2 Main Results

We conduct a comprehensive comparison of the models' performance by evaluating key metrics, including precision, recall, and F1 score, on the CULane dataset. The results are summarized in Table 1.

Model	Precision	$\Delta_{\text{Pre.}}$	Recall	$\Delta_{\text{Rec.}}$	F1	Δ_{F1}
Baseline	86.26	/	72.46	/	78.76	/
Proposed	86.39	$\uparrow$ 0.13	72.97	$\uparrow$ 0.51	79.11	$\uparrow$ 0.35

Table 1: Comparison of precision, recall, and F1 score between the baseline and the proposed model, evaluated on the CULane dataset.

The experimental results clearly demonstrate that the proposed method achieves superior performance compared to the baseline model, as evidenced by the increases in precision, recall, and F1 score. These improvements validate the efficacy of integrating PAFPN into CLRNet for enhanced feature representation.

4.3 FPS Results

To evaluate the computational efficiency of the proposed modifications, we measure the frames per second (FPS) during inference using dummy input data. The results, averaged over 30 iterations, are presented in Table 2.

Model	FPS	Δ_{FPS}
Baseline	309	/
Proposed	289	$\downarrow$ 6.5%

Table 2: Comparison of FPS between the baseline and the proposed model on a single NVIDIA RTX 4090 GPU.

The FPS comparison indicates that the inclusion of PAFPN introduces only a minimal reduction in computational efficiency, with a 6.5% decrease in FPS relative to the baseline model. This trade-off is considered acceptable given the significant performance gains achieved.

5 Future Work

To improve the generalizing ability, we propose the following future work to be finished.

- **Data Diversity Augmentation:** Increasing the diversity of training data by incorporating more variations in weather, lighting conditions, and road types can improve the model's robustness.
- **Cross-Domain Transfer Learning:** Leveraging transfer learning could help improve the model's ability to generalize across different domains and datasets.
- **Model Compression and Lightweight Architectures:** Balance computation cost and generalization.

6 Conclusion

The findings of this study illustrate that modifying the feature extraction network, specifically by replacing the FPN with PAFPN, can effectively enhance the performance of the CLRNet model with only a slight impact on computational speed. As a direction for future research, efforts could be directed toward optimizing the PAFPN architecture to further reduce its computational overhead while maintaining or even enhancing the observed performance improvements. Such optimizations could enable the proposed method to achieve similar levels of accuracy and robustness without compromising the inference speed.

References

[1] P. L. X. W. Xingang Pan, Jianping Shi and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI Conference on Artificial Intelligence (AAAI)*, February 2018.

[2] TuSimple, "Tusimple/tusimple-benchmark: Download datasets and ground truths."

[3] P. L. X. W. Xingang Pan, Jianping Shi and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *AAAI Conference on Artificial Intelligence (AAAI)*, February 2018.

[4] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai, "Resa: Recurrent feature-shift aggregator for lane detection," 2021.

[5] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, and Z. Li, "Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending," 2020.

[6] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[7] Z. Qin, H. Wang, and X. Li, "Ultra fast structure-aware deep lane detection," 2020.

[8] L. Liu, X. Chen, S. Zhu, and P. Tan, "Condlanenet: a top-to-down lane detection framework based on conditional convolution," 2023.

[9] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai, and X. He, "Clrnet: Cross layer refinement network for lane detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 898–907, June 2022.

[10] H. Honda and Y. Uchida, "Clrernet: Improving confidence of lane detection with laneiou," 2023.

[11] J. Han, X. Deng, X. Cai, Z. Yang, H. Xu, C. Xu, and X. Liang, "Laneformer: Object-aware row-column transformers for lane detection," 2022.

[12] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," 2020.

[13] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," 2020.

[14] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, and J. Yan, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," in *European Conference on Computer Vision (ECCV)*, 2022.

[15] P. Z. W. Z. J. M. J. W. Yuliang Guo, Guang Chen and T. E. Choe, "Gen-lanenet: A generalized and scalable approach for 3d lane detection," 2020.

[16] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018.

A Experiment Setup Detail

Item	Value
Optimizer	AdamW
Learning Rate	0.6e-3
Weight Decay	0.01
Batch Size	48
Epochs	15

Table 3: Experiment Setup Detail. Same for baseline and our method.