IFPNet: Integrated Feature Pyramid Network with Fusion Factor for Lane Detection

Zinan Lv[1] Dong Han[1] Wenzhe Wang[2,3] Cheng Chen[1]

[1]Harbin Engineering University [2]Zhejiang University [3]Westlake University

Abstract

Lane detection is a basic but challenging task in autonomous driving systems. With a combination of high-level and low-level information, early studies of lane detection have achieved promising results in some scenes. However, achieving better performance is still an urgent need for complex and diverse road conditions. We assume that learning and balancing the finer-scale features and global semantics is one of the keys to improving lane detection performance under these road conditions. In this paper, we propose an integrated feature pyramid network with fusion factor (IFPNet) for better hierarchical information learning and balancing, where a novel FPN structure named Integrated Feature Pyramid (IFP) is proposed for better hierarchical information integration. Classification Fusion Factor (CFF) is also utilized for the balance of hierarchical information. Moreover, we design the regression IoU (RIoU) loss for curve regression, which measures the overlap of the predicted and ground truth lane lines more effectively. We conduct experiments on three benchmark datasets of lane detection and achieve state-of-the-art results with high accuracy and efficiency.

1. Introduction

Lane detection is a fundamental but challenging task in autonomous driving systems, which can help vehicles get their relative location and respond to emergencies in real-time. Traditional autonomous driving techniques rely on multi-sensor fusion, which requires expensive inertial guidance components and LIDAR [29]. But with the help of lane detection, vehicles can automatically detect lane lines from images captured only by front-mounted cameras.

Early studies of lane detection focus on the utilization of manually extracted features based on priors such as color and structure [12, 13, 33]. After pre-processing the possible lane lines obtained from the extracted features, methods like Hough transform [18] and Kalman filter [40] are used to filter out unreal or densely predicted lines to output the final lane lines. These methods commonly get limited results in various road environments. Recently, with the development of CNN, segmentation-based methods [24, 41, 43] are proposed for lane detection. However, treating lane detection as a common segmentation task ignores lane lines' narrow, continuous, and structured characteristics. To alleviate this issue, anchor-based [14, 28, 31] and curve-based [4, 20, 35] methods with faster speed are proposed to achieve better detection results.

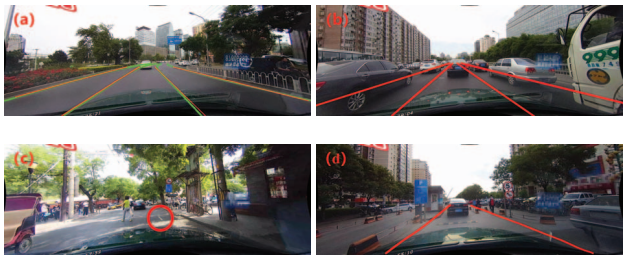

Figure 1. Examples of complex and diverse road conditions in the lane detection task. (a) The case where the curve at the end of the field of view can be easily ignored and mistaken as a straightaway by detection methods. (b) Crowded scenes during peak periods where most of the lane lines are hidden. (c) The shadows of trees make light spots on the ground look like lane lines. (d) The case where there is no obvious white or yellow line on the ground. (a) and (b) indicate complex road conditions in the lane detection task whereas (c) and (d) indicate diverse road conditions in the task.

Although better results have been obtained, further improving the accuracy of lane detection is still a vital task. As shown in Fig. 1, recently proposed lane detection methods may still suffer from unsatisfactory results in some complex and diverse road conditions. In Fig. 1(a), a curve is in front of the vehicle and at the end of the field of view. It can be easily ignored and mistaken as a straightaway by lane detection methods, leading to less reaction time for the vehicle to turn. A scene of rush hour or traffic jams is illustrated in Fig. 1(b), where most of the lane lines are hidden

by other vehicles. The above two examples indicate complex road conditions, where learning finer-scale features is essential for subtle lane line detection. For diverse road conditions, in Fig. 1(c), the shadows of trees cause dashed-like light spots on the ground, misleading the detection methods about the appearance of a lane line there. In Fig. 1(d), the lane lines are not common white or yellow lines but are composed of protrusions on the edge of the road, manually set railings, and so on. For the above two examples, besides the finer-scale features for subtle lane line detection, global semantics is also essential during the process of lane line feature learning and refinement. According to the four examples in Fig. 1, finer-scale features and global semantics are two indispensable parts of the lane detection task. However, recent works usually failed on these complex and diverse road conditions, indicating the limited ability of finer-scale features and global semantic extraction. CLRNet [44] proposes a network that refines features from high-level to low-level to exploit contextual information, but hierarchical information can be learned and balanced.

In this paper, we propose an integrated feature pyramid with a fusion factor (IFPNet) for lane detection, which can achieve better hierarchical information learning and balancing. Specifically, a novel FPN [16] structure named Integrated Feature Pyramid (IFP) is proposed to improve hierarchical information learning. Inspired by [8], a fusion factor module named Classification Fusion Factor (CFF) is introduced for hierarchical information balancing. Unlike the traditional FPN structure [16], our proposed IFP focuses on integrating finer-scale features and global semantics in the lane detection task. Moreover, to make the predicted curves fit the ground truth lane lines better, we refer to the IoU loss in the object detection task and propose a regression IoU (RIoU) loss, which measures the overlap of the predicted and ground truth lane lines in a more effective way. Experimental results on three lane detection benchmark datasets demonstrate the effectiveness of our method. The main contributions of this paper can be summarized as follows:

- We propose an integrated feature pyramid with fusion factor (IFPNet) for lane detection, where a novel FPN structure named Integrated Feature Pyramid (IFP) is proposed for better hierarchical information learning. The CFF module is applied to the IFP for better hierarchical information balancing. The proposed IFPNet is a plug-and-play module that can be applied on different backbones in different applications.
- For better curve regression, we propose a regression IoU (RIoU) loss to measure the overlap of the predicted and ground truth lane lines.
- The state-of-the-art results have been achieved on lane detection benchmark datasets, demonstrating the effectiveness of our proposed IFPNet. Ablation studies are also carried out to evaluate the effectiveness of each part of the IFPNet.

2. Related Works

Deep learning-based lane detection methods can be divided into three mainstream categories corresponding to three kinds of lane representation, which are segmentation-based methods, curve-based methods, and anchor-based methods, respectively.

2.1. Segmentation-based methods

Segmentation-based methods [7, 10, 41, 42] treat lane detection as a per-pixel segmentation task. Since CNN was introduced into the lane detection task, this kind of method has been continuously refined. The SCNN method [24] significantly improves the performance of lane detection tasks compared to traditional methods, but the computational speed of the method is relatively slow for real-time applications. LaneNet [23] proposes an end-to-end instance segmentation pipeline, but it needs two separate networks to cluster and obtain the structure of lane lines, which is also time-consuming. Realizing the importance of global semantics, RESA [43] proposes a feature aggregation module to gather global features. It achieves real-time application but still suffers from unsatisfactory results in complex and diverse road conditions. Since the segmentation-based methods cannot bypass the pixel-wise prediction step, irrelevant computation on the background area is inevitable, leading to GPU resources and time consumption and the degradation of real-time performance.

2.2. Curve-based methods

The output of curve-based methods [4, 20, 35] is parametric lines composed of curve equations (*e.g.*, $x = ay^3 + by^2 + cy + d$, where (x, y) represents the coordinate of pixels and a, b, c, and d denote the parameters of a line). PolyLaneNet [32] proposes an end-to-end deep polynomial regression method that outputs a polynomial to represent each lane marker in an input image. BézierLaneNet [5] proposes a novel Bézier curve-based deep lane detector, which can model the geometric shape of lane lines effectively. With the development of Transformer [36] on computer vision tasks, some studies have introduced Transformer into the lane detection task [3, 20] for better global semantics learning. Compared to segmentation-based methods, curve-based methods commonly have fewer parameters with faster inference speeds. However, since they are sensitive to the predicted parameters [44], their performance is not guaranteed, they may struggle especially on large and complex datasets.

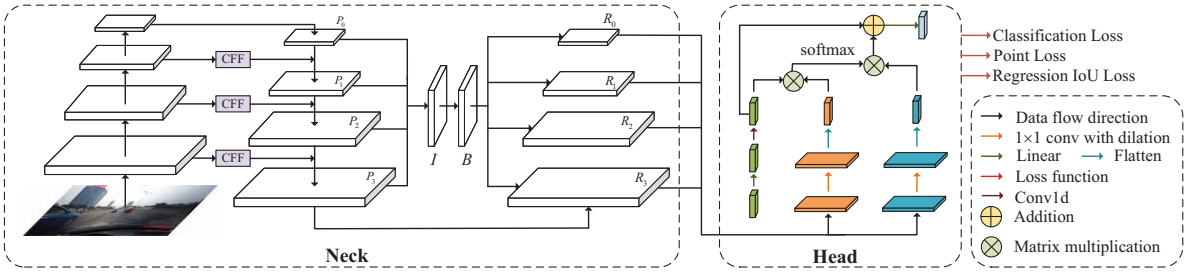

Figure 2. The overall architecture of our proposed IFPNet, where "Neck" and "Head" denote components in our proposed FPN (IFP). The input of the neck of IFP is multi-scale features output from a backbone network. The Classification Fusion Factor module (CFF) is applied to balance global semantics and finer-scale features. The head of IFP is responsible for the generation of lane anchors, where an attention structure is utilized for better feature learning. Regression IoU loss helps improve the regression of lane lines during training.

2.3. Anchor-based methods

Anchor-based methods [9, 14, 30, 31, 37] are currently a popular direction of the lane detection task. UFLD [28] is the first to propose the row-wise strategy, where a lightweight backbone and a streamlined network enable extremely fast inference speed. However, in some complex scenes (*e.g.*, crowded, shadow, and night), its accuracy can be obviously reduced. Similar works [19, 42] further explore row-wise methods and achieve better performance, but they still get limited results on the prediction of the start and end points of each lane and the regression of curves. On the other hand, for better utilization of contextual information, LaneATT [31] introduces an anchor-based attention mechanism to aggregate global semantics. Similar to the former work, CLRNet [44] proposes a network that refines features from high-level to low-level. It achieves better results than all the above methods.

3. Method

The overall architecture of our proposed IFPNet is illustrated in Fig. 2. In the following of this section, we will first introduce the detailed information of our proposed FPN structure named IFP, and then show the definition of the proposed regression IoU (RIoU) loss.

3.1. Integrated Feature Pyramid

3.1.1 Classification Fusion Factor Module

FPN [16] is a structure that combines high-level information with low-level information. Previous research [8] has demonstrated that the performance of FPN is affected by the fusion proportion between two adjacent feature layers, which can be defined in the following manner:

$$P_i = f_{lat}\left(P_i + \alpha \times f_{inter}\left(P_{i+1}\right)\right), \tag{1}$$

where P_i is the i-th layer of FPN, f_{lat} is the 1×1 convolution operation, α is the fusion proportion between the adjacent feature layers, and f_{inter} denotes the $2\times$ up-sampling operation. As shown in Fig. 3, the prediction results on different types of images in the lane detection task are affected by the fusion proportion, which is denoted as α in this paper. Besides, for different types of images, the optimal point corresponding to the peak of the F1-measure varies, thus using a fixed fusion proportion α hampers us from getting a better performance for up-sampling. Hence an image-classification-based method is urgently needed for calculating the optimal fusion factor for different types of images.

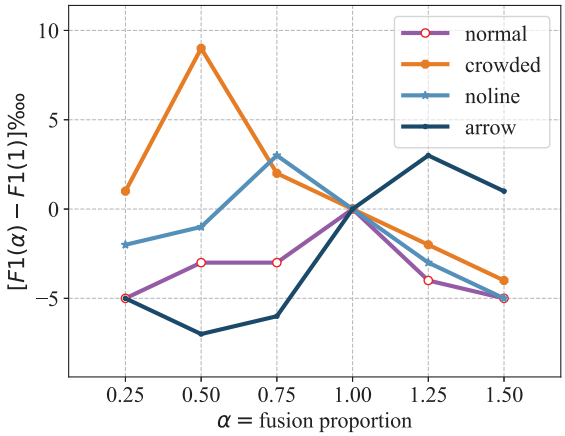

Figure 3. The relationships between F1-measure and different fusion proportions on different types of images, where the y-axis represents the improvement of F1-measure when α in the original FPN is changed from 1 to another value.

Different from the traditional FPN structure that simply adds the low-level location information and the high-level semantic information, we propose a Classification Fusion Factor module (CFF) to further integrate high-level and low-level information, which is more lightweight and efficient. The detailed structure of CFF is shown in Fig. 4. Before doing the up-sampling operation, the high-level fea-

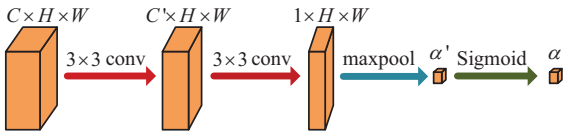

Figure 4. Structure of Classification Fusion Factor module (CFF). Convolution in the figure is 3×3. C' is an adjustable parameter in the hidden layer, α' represents $1 \times 1 \times 1$ matrix, and α denotes the classification fusion factor.

tures go through an unsupervised classification module CFF to get the optimal fusion factor α to integrate high-level information with low-level information more accurately and effectively.

3.1.2 Feature Integration

High-level features from the backbone usually have more global information while low-level features usually have more local information. The question is about how to integrate high-level and low-level information to get better performance. Inspired by former research [25, 27], we use the refinement feature pyramid here to balance the information from high-level to low-level features, which can help the head of the net to refine anchors accurately.

Since the scales of FPN's output are different, we have to unify the scales before integrating high and low-scale features. The scale should be moderate in order to prevent valid information lost or increase the computational effort. After experimentation, we found that reshaping the output layer to the second layer reaches the optimal while balancing the local and global information, represented in Fig. 2 as P_1. All multi-scale features are reshaped to the shape of P_1 with no extra parameters to learn in order to save the computation resources. Then we take averages for these same scale features to get the balanced information of different levels of features, which can be mannered as follows:

$$I = \frac{1}{N} \sum_{i=l_{min}}^{l_{max}} C_i^{'}, \tag{2}$$

where I is the integrated multi-scale features, N is the number of the used backbone layers which begin from l_{min} to l_{max}, and $C_i^{'}$ is the feature map after scale unification from up-sampling or down-sampling.

After the initial fusion, the global and local information has been condensed into a feature map (I in Fig. 2). However, in some complex scenes, global lane features are harder to capture, and precise lane location relies on long-distance contextual information. In order to further capture the spatial dependencies of any two locations in the feature map and obtain long-range context dependency information, we employ a self-attentive mechanism [36] to further integrate the global contextual information. Numerous studies have been conducted on self-attention mechanisms in the computer vision field [6, 11, 15] during the past five years. We found that both non-local [38] and CBAM [39] work well in balancing the local and global information for lane detection. The default module here is non-local, where CBAM is also supported. The last step of IFP is the reshaping of the refined layer B with up-sampling or down-sampling and then adding them to the output of CFF with a learnable parameter σ, which can be represented in the following manner:

$$R_i = (1 - \sigma)\, C_i + \sigma R_i^{'}. \tag{3}$$

3.2. Attention Lane Detection

3.2.1 Lane Anchor

When given a series of input images R_0, R_1, R_2, R_3, the goal of the head is to output a series of lines composed of key points on the line, which is defined as follows:

$$l_k = \left\{ \left(x_1^{(k)}, y_1^{(k)} \right), \cdots, \left(x_n^{(k)}, y_n^{(k)} \right) \right\}, l_k \subseteq L \tag{4}$$

where l_k means the k-th lane of the image, n means the number of key points on the lane, and L means the set of lane lines in the image. All the key points of the line are equally interval sampling by the y-axis, which is consistent with the real situation of the lane line. To get the line set, first, we get an anchor that expresses the basic information about the lane line, which can be defined as:

$$A_i = \{\hat{p}_i, x_s, y_s, \theta, Len, y_1, y_2, \ldots, y_n\}, \tag{5}$$

where $\hat{p}_i$ indicates the probability that the i-th line is whether a lane line or background, (x_s, y_s, θ) represents a straight line that the start point of the line is (x_s, y_s) and the regression angel of the line is θ. Len means the total length of the line. y_1 to y_n means the horizontal offset distance of the accurate points from the regression points.

3.2.2 Attention Structure

Inspired by the structure of the transformer, the main structure of the head of the net adopts the refinement structure, where the prior estimation of line anchors is generated by positional embeddings and is refined by feature maps of four scales from IFP in Sec. 3.1. The order of the refinement is from high-level features to low-level features, which can help the network to get the global lane information initially and then correct the lane lines based on the local information.

The detailed structure of the refinement module adopts the attention module. Different from the traditional self-attention module [36], we use dilated convolution to calculate the key and value of the self-attention. It is experimentally demonstrated that the use of dilation to increase the

receptive field helps the network to obtain more global information to reduce the false detection of lane lines due to local information such as light spots. The output matrix of this module can be calculated in the following manner:

$$A_i = Softmax\left(\frac{A_{i-1}K^T}{\sqrt{d_k}}\right)V + A_{i-1}, \tag{6}$$

where K and V are the output of IFP after stretching, and A_{i-1} is the anchor refined by the last layer (A_0 is the prior estimation of the line anchor).

3.3. Loss

Classification Loss As shown in Sec. 3.2.1, $\hat{p}_i$ denotes the probability of the i-th line is a lane line calculated by the linear layer. To better distinguish between the lane lines and background, we use classification loss, which is defined as:

$$\ell_{cls} = \sum_{i=1}^{R}\sum_{j=1}^{L}\lambda_{cls}\mathcal{L}\left(\hat{p}_{i,j}, G_{i,j}\right), \tag{7}$$

where R is the number of refinement layers, L is the number of lines in the ground truth, λ_{cls} is the weight of different refinement layers, $\mathcal{L}$ is the cross-entropy loss, and $G_{i,j}$ is the ground truth of whether the line is a lane line or not.

Point Loss The start and end points of the lane lines are crucial for the precise positioning of the lane lines. To find the exact start of the lane, we propose point loss, which corresponds to:

$$\ell_{point} = \sum_{i=1}^{R}\sum_{j=1}^{L}\left\|\hat{S}_{i,j} - S_{i,j}\right\|_1, \tag{8}$$

where $\hat{S}_{i,j}$ is the start point of predicted line, $S_{i,j}$ is the start point of ground truth line, and $\|\cdot\|_1$ is $L1$ norm.

RIoU Loss The regression of the curve is crucial for the precise positioning of the lane lines. And the regression of the curve is composed of key points, *i.e.*, the offset of the row-wise position. We propose the regression IoU loss (RIoU Loss) which represents the overlap of curves, which can be written as follows:

$$\ell_{RIoU} = 1 - IoU + \frac{\lambda_{RIoU}}{n}\left|\hat{y}_i - y_i\right|, \tag{9}$$

where IoU is the sum of the overlapped pixels between the predicted lane line and the ground truth lane line on the row of the line, n is the total number of key points of the lane line, and λ_{RIoU} is the loss coefficient that combines IoU and regression distance.

Loss Aggregation The training loss of the net is composed of three parts, classification loss, point loss, and RIoU loss. The overall loss function ℓ can be written as:

$$\ell = \lambda_1\ell_{cls} + \lambda_2\ell_{point} + \lambda_3\ell_{RIoU}, \tag{10}$$

where λ_1, λ_2, and λ_3 are coefficients of classification, point, and RIoU loss, respectively.

Dataset	Train / Val / Test (K)	Scene
LLAMAS [2]	5.8 / 2.1 / 2.1	Highway
Tusimple [34]	3.3 / 0.4 / 2.8	Highway
CULane [24]	88.9 / 9.7 / 34.7	Urban&Highway

Table 1. Detailed information of the datasets.

4. Experiments and Results

4.1. Datasets

To verify the effectiveness of our proposed method, we conduct experiments on three lane detection benchmark datasets: CULane [24], Tusimple [34], and LLAMAS [2]. CULane is a widely used large dataset on lane detection including eight hard-to-detect conditions. Tusimple is a commonly used dataset with images captured on the highway, with clear weather and clear lane lines. LLAMAS is a recently released dataset captured on the highway. Details of the datasets are shown in Table 1.

4.2. Evaluation Metrics

For the CULane [24] and LLAMAS [2], we adopt the F1-measure proposed by SCNN [24] as the evaluation metrics. Intersection-over-Union (IoU) between the predicted lane line of the net and Ground Truth (GT) label is calculated to judge whether a sample is True Positive (TP), False Positive (FP), or False Negative (FN). The F1 score is calculated in the following manner:

$$Precision = \frac{TP}{TP+FP}, Recall = \frac{TP}{TP+FN}, \tag{11}$$

$$F1 = \frac{2\times Precision\times Recall}{Precision + Recall}. \tag{12}$$

For rigorous concerns, we also calculate mF1 as the evaluation metrics compared with the localization performance of algorithms according to the COCO dataset [17], which can be written as:

$$mF1 = \frac{F1@50 + F1@55 + \cdots + F1@95}{10}, \tag{13}$$

where $F1@50$, $F1@55$, …, $F1@95$ represent the calculated results of F1 where IoU thresholds are 0.5, 0.55, …, 0.95 respectively.

For Tusimple [34], the evaluation metrics are composed of three official indicators named accuracy, false positive rate (FPR), and false negative rate (FNR). The method to calculate accuracy is shown in the following manner:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}, \tag{14}$$

where C_{clip} is the number of correct points and S_{clip} is the number of Ground Truth (GT) points in an input image. If

Method	Backbone	mF1	total	normal	crowded	hlight	shadow	noline	arrow	curve	cross	night	FPS	GFlops
SCNN [24]	VGG16	38.84	71.60	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10	7.5	328.4
FastDraw [26]	ResNet50	-	-	85.90	63.60	57.00	69.90	40.60	79.40	65.20	7013	57.80	90.3	-
UFLD [28]	ResNet18	38.94	68.40	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10	**341**	**8.4**
UFLD [28]	ResNet34	-	72.30	90.70	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70	184	-
RESA [43]	ResNet34	-	74.50	91.90	72.40	66.50	72.00	46.30	88.10	68.60	1896	69.80	51	-
RESA [43]	ResNet50	-	75.30	92.10	73.10	69.20	72.80	47.70	88.30	70.30	1503	69.90	39	-
LaneATT [31]	ResNet18	47.35	75.13	91.17	72.71	65.82	68.03	49.13	87.82	63.75	1020	68.58	176	9.3
LaneATT [31]	ResNet34	49.57	76.68	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72	145	18.0
LaneATT [31]	ResNet122	51.48	77.02	91.74	76.16	69.47	76.31	50.46	86.29	64.05	1264	70.81	31	70.5
SGNet [30]	ResNet18	-	76.12	91.42	74.05	66.89	72.17	50.16	87.13	67.02	1164	70.67	135	-
SGNet [30]	ResNet34	-	77.27	92.07	75.41	67.75	74.31	50.90	87.97	69.65	1373	72.69	116	-
CondLane [19]	ResNet18	51.84	78.14	92.87	75.79	70.72	80.01	52.39	89.37	72.40	1364	73.23	201	10.2
CondLane [19]	ResNet34	53.11	78.74	93.38	77.14	71.17	79.93	51.85	89.89	73.88	1387	73.92	140	19.6
CondLane [19]	ResNet101	54.83	79.48	93.47	77.44	70.93	80.91	54.13	90.16	75.21	1201	74.80	56	44.8
GANet [37]	ResNet18	-	78.79	93.24	77.16	71.24	77.88	53.59	89.62	75.92	1240	72.75	164	-
GANet [37]	ResNet34	-	79.39	93.73	77.92	71.64	79.49	52.63	90.37	76.32	1368	73.67	151	-
GANet [37]	ResNet101	-	79.63	93.67	78.66	71.82	78.32	53.38	89.86	**77.37**	1352	73.85	67	-
CLRNet [44]	ResNet18	55.23	79.58	93.30	78.33	73.71	79.66	53.14	90.25	71.56	1321	75.11	226	11.9
CLRNet [44]	ResNet34	55.14	79.73	93.49	78.06	74.57	79.92	54.01	90.59	72.77	1216	75.02	184	21.5
CLRNet [44]	ResNet101	55.55	80.13	**93.85**	78.78	72.49	82.33	**54.50**	89.79	75.57	1262	75.51	170	42.9
IFPNet	ResNet18	55.43	79.95	93.57	78.13	**75.78**	81.74	53.39	90.50	71.69	**1017**	75.54	208	14.2
IFPNet	ResNet34	55.54	79.80	93.57	78.52	73.33	80.61	53.75	90.34	72.92	1182	74.97	171	23.6
IFPNet	ResNet101	**56.32**	**80.33**	93.58	**78.94**	75.06	**82.50**	54.21	**90.68**	73.26	1068	**75.81**	105	44.2

Table 2. State-of-the-art results of recently proposed methods on CULane. In order to test speed in the same environment, we remeasure FPS on the same machine with an RTX3090 GPU through open-source code.

the accuracy of a predicted lane is greater than 85%, it will be considered a True Positive (TP). The F1 score is also used during the evaluation.

4.3. Implementation Details

In the experiments, we adopt ResNet as the pre-trained backbone for our model. For all the proposed models, the number of the used backbone layers N layers is set to 4. The length of the lane anchor n is set to 72. All input images are resized to 800×320 pixels for training or testing. The loss coefficients in Sec. 3.3 are set 2.0, 0.2, 2.0 corresponding to λ_1, λ_2, and λ_3. The training epochs for CULane, Tusimple, and LLAMAS are set to 20, 90, and 20, respectively. In the optimizing process, we use AdamW [22] and cosine decay learning rate strategy [21] with an initial learning rate of 6e-4. We train our model with a batch size of 32 on CULane, Tusimple, and LLAMAS for 20, 90, and 20 epochs, respectively. All the experiments are conducted on an NVIDIA RTX3090 GPU.

4.4. Results

Results on CULane Comparisons of results of recently proposed methods and our work on the CULane dataset are shown in Table 2. Our work achieves state-of-the-art results on both mF1 and the total F1 score. Our method achieves the best performance in six of the eight difficult scenarios. In particular, our proposed IFPNet achieves the best results on "hlight", "shadow" and "night", showing our method can adapt to complex environments with different light conditions. Among them, "hlight" is 1.21% higher than the previous best result.

The visualization results on the benchmark dataset CULane of our method and previous best-performance methods are shown in Fig. 5. ResNet18 is taken as the backbone of each method to compare the visualization results on the same scale. CondLaneNet solves the problem of fork lines and dense lines but the continuity of the line is not well. CLRNet only refines the anchor from high-level to low-level instead of integrating them deeply, so it is easy to miss lane lines under complex environments. Our method works well in high brightness and dim light. Moreover, our method is able to regress to the lane line more accurately, both during the day and at night.

Results on Tusimple Comparisons of results of recently proposed methods and our work on the Tusimple dataset are shown in Table 2. Our method achieves new state-of-art results on F1, False Negative Rate (FN), and accuracy, demonstrating that our method can be adapted to both complex urban environments and simple highway scenarios. Because the dataset is relatively simple (lane features are obvious), the results are close.

Results on LLAMAS Comparisons of results between recently proposed methods and our work on the LLAMAS dataset are shown in Table 4. Our IFPNet performs well on the dataset and we achieve state-of-art results on both mF1 and F1@75 scores. Due to the simplicity of the LLAMAS dataset (on the highway with obvious features), the results are close.

(a) Dazzle light (b) Crowded (c) Curve at day (d) Night (e) Curve at night

Figure 5. The visualization results of Ground Truth (GT), CondLaneNet [19] (CondLane in the figure), CLRNet [44], and our method IFPNet on the benchmark dataset CULane [24]. The results are generated with the same backbone ResNet18.

Method	Backbone	F1	Acc	FP	FN
SCNN [24]	VGG16	95.97	96.53	6.17	1.80
RESA [43]	ResNet34	96.93	96.82	3.63	2.48
FastDraw [26]	ResNet50	93.92	95.20	7.60	4.50
UFLD [28]	ResNet18	87.87	95.82	19.05	3.92
UFLD [28]	ResNet34	88.02	95.86	18.91	3.75
PolyLaneNet [32]	EfficientNetB0	90.62	93.36	9.42	9.33
LaneATT [31]	ResNet18	96.71	95.57	3.56	3.01
LaneATT [31]	ResNet34	96.77	95.63	3.53	2.92
LaneATT [31]	ResNet122	96.06	96.10	5.64	2.17
CondLaneNet [19]	ResNet18	97.01	95.48	2.18	3.80
CondLaneNet [19]	ResNet34	96.98	95.37	2.20	3.82
CondLaneNet [19]	ResNet101	97.24	96.54	**2.01**	3.50
CLRNet [44]	ResNet18	97.89	96.84	2.28	1.92
CLRNet [44]	ResNet34	97.82	96.87	2.27	2.08
CLRNet [44]	ResNet101	97.62	96.83	2.37	2.38
IFPNet	ResNet18	97.83	96.75	2.07	2.27
IFPNet	ResNet34	**97.93**	96.73	2.34	1.78
IFPNet	ResNet101	97.65	**96.94**	2.95	**1.71**

Table 3. State-of-the-art results on Tusimple.

Method	Backbone	mF1	F1@50	F1@75	GFlops
LaneATT [31]	ResNet18	69.22	94.64	82.36	**9.3**
LaneATT [31]	ResNet34	69.63	94.96	82.79	18.0
LaneATT [31]	ResNet122	70.80	95.17	84.01	70.5
LaneAF [1]	DLA34	69.31	96.90	84.71	23.6
CLRNet [44]	ResNet18	71.61	96.96	85.59	11.9
CLRNet [44]	ResNet101	71.21	**97.16**	85.33	18.5
IFPNet	ResNet18	70.62	96.61	84.62	14.3
IFPNet	ResNet101	**71.63**	97.00	**85.66**	44.0

Table 4. State-of-the-art results on LLAMAS.

4.5. Ablation Study

To further demonstrate the role of each module in our net for lane detection, we conducted ablation experiments on the benchmark dataset CULane using the same backbone ResNet18.

Overall Ablation Study We use UFLD [28] as the baseline, where we gradually add Attention Head, CFF, IFP, and RIoU on it. The results of the overall ablation study are shown in Table 5. The first row shows the result of the baseline, which is consistent with the result of the open-source code of UFLD. The attention head improves the F1 score from 68.40 to 78.56, which is the most effective module of the net. As the module is gradually added, the F1 score is increasing simultaneously. Specifically, CFF and IFP improve the F1 score by 0.57% and 0.91% respectively. The combination of CFF and IFP is more effective for the improvement of the F1 score, which improves the score by 1.18%. The usage of RIoU further improves the score by 0.21%, which helps the regression of the lane line.

Attention Head	CFF	IFP	RIoU	F1 score
				68.40
✓				78.56
✓	✓			79.13
✓		✓		79.47
✓	✓	✓		79.74
✓	✓	✓	✓	**79.95**

Table 5. Results of the overall ablation study on CULane using the same backbone ResNet18. The baseline of our study is UFLD [28], where all the modules are added on it.

The visualization results of the module CFF and IFP are shown in Fig. 6. Both the CFF module and IFP structure can help the detection of lane lines, but the combination of the two modules can use the finer-scale features and the global semantics more effectively. For example, as can see from Fig. 4.5(e), when only CFF is added to the net, it can not detect all the finer-scale features of the image. And when only IFP is added to the net, it neglects the global semantics of the lane which makes it mistake the arrow as the lane line. The combination of CFF and IFP can predict lane lines more accurately. Our IFP can be used in other lane detection

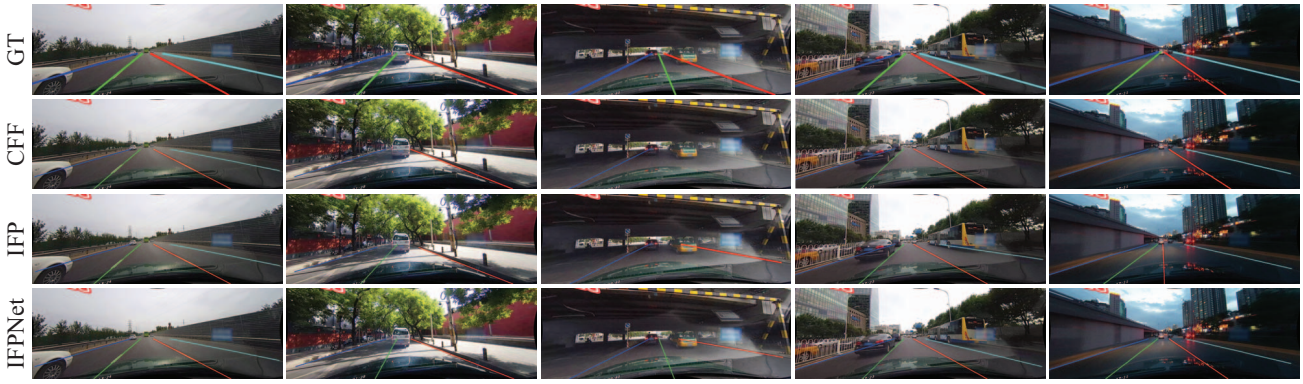

Figure 6. The visualization results of the module CFF and IFP. The first row denotes Ground Truth (GT). The second row to the fourth row represents the results that only add the CFF module, results that only add the IFP module and results that add CFF and IFP modules.

Method	Dilation shape	Padding shape	F1 score
IFPNet	-	-	79.72
IFPNet	3×3	-	79.80
IFPNet	3×3	3×3	79.67
IFPNet	5×5	-	79.78
IFPNet	5×5	5×5	**79.95**
IFPNet	7×7	-	79.34
IFPNet	7×7	5×5	79.49

Table 6. Results of the ablation study on attention head on CULane based on the same backbone ResNet18.

methods, results can be seen in the appendix of our work.

Ablation Study on Attention Head As mentioned in Sec. 3.2.2, we use dilated convolution to calculate the key and value of self-attention. The result of different shapes of dilation and padding is shown in Table 6. The F1 score increases with the dilation shape from 3×3 to 5×5 with the same shape padding to keep the original shape of the input image, which demonstrates the validity of dilation in the attention head.

Ablation Study on RIoU The result of ablation studies of RIoU is shown in Table 7. We use smooth-l_1, LIoU [44], and RIoU as different loss modules of the study. The only change in each study is the weight of each loss. Tradition smooth-l_1 loss does not fully utilize the structural features of the lane lines. LIoU loss proposed by CLRNet [44] takes the lane line as a whole unit which is helpful for the performance of lane detection. By contrast, RIoU loss is more helpful for the regression of lane lines on the large, medium, and large types of IFP, which demonstrates the efficiency of our proposed RIoU loss. To further verify the effectiveness of RIoU, we add RIoU loss to LaneATT [31] and CLRNet [44], where we achieve improved performance. Results can be seen in the Appendix of our work.

Loss	Weight	ResNet18	ResNet34	ResNet101
smooth-l_1	0.1	79.56	-	-
smooth-l_1	0.5	79.46	79.63	79.71
smooth-l_1	1.0	79.53	79.54	-
LIoU	1	79.68	-	-
LIoU	2	79.74	79.61	79.95
LIoU	4	79.69	-	79.86
RIoU	0.1	79.76	79.65	80.11
RIoU	0.5	79.90	**79.85**	80.15
RIoU	1.0	**79.95**	79.80	80.19
RIoU	2.0	79.89	79.79	**80.33**

Table 7. Results of the ablation study on RIoU. The results are based on IFP on the benchmark dataset CULane. The only change in each row is the weight of each loss.

5. Conclusion

In this paper, we propose Integrated Feature Pyramid Network (IFPNet) based on the fusion factor for lane detection. To improve the accuracy of lane detection under complex and diverse road conditions like crowded scenes or uneven and insufficient light, we propose Classification Fusion Factor (CFF) and IFP to further integrate finer-scale features and global semantics. Moreover, we propose regression IoU (RIoU) to measure the overlap of the predicted and ground truth lane lines, which is helpful for the regression of curves. We test our method on three benchmark datasets including CULane, Tusimple, and LLAMAS, where we achieve state-of-the-art results with both high efficiency and high accuracy.

6. Acknowledgement

We thank Shiyu Zhang, Westlake University for helping with proofreading and grammar checking for this paper.

References

[1] Hala Abualsaud, Sean Liu, David B Lu, Kenny Situ, Akshay Rangesh, et al. Laneaf: Robust multi-lane detection with affinity fields. *IEEE Robotics and Automation Letters*, 6(4):7477–7484, 2021. 7

[2] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane markers using maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 5

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, et al. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020. 2

[4] Haoxin Chen, Mengmeng Wang, and Yong Liu. Bsnet: Lane detection via draw b-spline curves nearby. *arXiv preprint arXiv:2301.06910*, 2023. 1, 2

[5] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, et al. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022. 2

[6] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, et al. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 4

[7] Mohsen Ghafoorian, Cedric Nugteren, Nóra Baka, Olaf Booij, and Michael Hofmann. El-gan: Embedding loss driven generative adversarial networks for lane detection. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[8] Yuqi Gong, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, et al. Effective fusion factor in fpn for tiny object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1160–1168, 2021. 2, 3

[9] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-region affinity distillation for road marking segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12486–12495, 2020. 3

[10] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1013–1021, 2019. 2

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 4

[12] Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo. Multi-lane detection in urban driving environments using conditional random fields. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 1297–1302. IEEE, 2013. 1

[13] Heechul Jung, Junggon Min, and Junmo Kim. An efficient lane detection algorithm for lane departure detection. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 976–981. IEEE, 2013. 1

[14] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258, 2019. 1, 3

[15] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019. 4

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 2, 3

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, et al. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[18] Guoliang Liu, Florentin Wörgötter, and Irene Markelić. Combining statistical hough transform and particle filter for robust lane detection and tracking. In *2010 IEEE Intelligent Vehicles Symposium*, pages 993–997. IEEE, 2010. 1

[19] Lizhe Liu, Xiaohao Chen, Siyu Zhu, and Ping Tan. CondLaneNet: a top-to-down lane detection framework based on conditional convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3773–3782, 2021. 3, 6, 7

[20] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3694–3702, 2021. 1, 2

[21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 286–291. IEEE, 2018. 2

[24] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 1, 2, 5, 6, 7

[25] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, et al. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2019. 4

[26] Jonah 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*, pages 11582–11591, 2019. 6, 7

[27] Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10213–10224, 2021. 4

[28] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 276–291. Springer, 2020. 1, 3, 6, 7

[29] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5692–5698. IEEE, 2021. 1

[30] Jinming Su, Chao Chen, Ke Zhang, Junfeng Luo, Xiaoming Wei, et al. Structure guided lane detection. *arXiv preprint arXiv:2105.05403*, 2021. 3, 6

[31] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, et al. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 294–302, 2021. 1, 3, 6, 7, 8

[32] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, et al. Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE, 2021. 2, 7

[33] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, and Keqiang Li. A novel curve lane detection based on improved river flow and ransa. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 133–138. IEEE, 2014. 1

[34] Tusimple. Tusimple benchmark. https://github.com/TuSimple/tusimple-benchmark/, Accessed September, 2020. 5

[35] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 4

[37] Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, et al. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1392–1401, 2022. 3, 6

[38] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 4

[39] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 4

[40] Pei-Chen Wu, Chin-Yu Chang, and Chang Hong Lin. Lane-mark extraction for automobiles under complex conditions. *Pattern Recognition*, 47(8):2756–2767, 2014. 1

[41] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, et al. Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 689–704. Springer, 2020. 1, 2

[42] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, et al. End-to-end lane marker detection via row-wise classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1006–1007, 2020. 2, 3

[43] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, et al. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3547–3554, 2021. 1, 2, 6, 7

[44] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, et al. CLRNet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 898–907, 2022. 2, 3, 6, 7, 8