DILane: Dynamic Instance-Aware Network for Lane Detection

Zhengyun Cheng, Guanwen Zhang✉, Changhao Wang, and Wei Zhou

Northwestern Polytechnical University, Xi'an, China
guanwen.zh@nwpu.edu.cn

Abstract. Lane detection is a challenging task in computer vision and a critical technology in autonomous driving. The task requires the prediction of the topology of lane lines in complex scenarios; moreover, different types and instances of lane lines need to be distinguished. Most existing studies are based only on a single-level feature map extracted by deep neural networks. However, both high-level and low-level features are important for lane detection, because lanes are easily affected by illumination and occlusion, *i.e.*, texture information is unavailable in non-visual evidence case; when the lanes are clearly visible, the curved and slender texture information plays a more important role in improving the detection accuracy. In this study, the proposed DILane utilizes both high-level and low-level features for accurate lane detection. First, in contrast to mainstream detection methods of predefined fixed-position anchors, we define learnable anchors to perform statistics of potential lane locations. Second, we propose a dynamic head aiming at leveraging low-level texture information to conditionally enhance high-level semantic features for each proposed instance. Finally, we present a self-attention module to gather global information in parallel, which remarkably improves detection accuracy. The experimental results on two mainstream public benchmarks demonstrate that our proposed method outperforms previous works with the F1 score of 79.43% for CULane and 97.80% for TuSimple dataset while achieving 148+ FPS.

Keywords: Lane Detection · Dynamic Head · Self-attention.

1 Introduction

With the recent development of artificial intelligence, various autonomous driving technologies [1] have achieved satisfying results. Lane detection is a fundamental perception task in autonomous driving with a wide range of applications, such as adaptive cruise control, lane keeping assistance, and high-definition mapping. To ensure the safety of autonomous driving and ensure basic driving between lane lines, lane detection requires high accurate and real-time. However, in actual driving scenarios, lane lines are considerably affected by complex environments such as extreme lighting conditions, occlusion, and lane line damage;

Code is available at https://github.com/CZY-Code/DILane

these are very common and make the task of lane detection challenging. The traditional lane detection methods [1, 2] usually rely on texture information of the lane line for feature extraction; then they obtain the topology structure of the lane line through post-processing regression. Yet, traditional methods lack robustness in complex real-world scenarios.

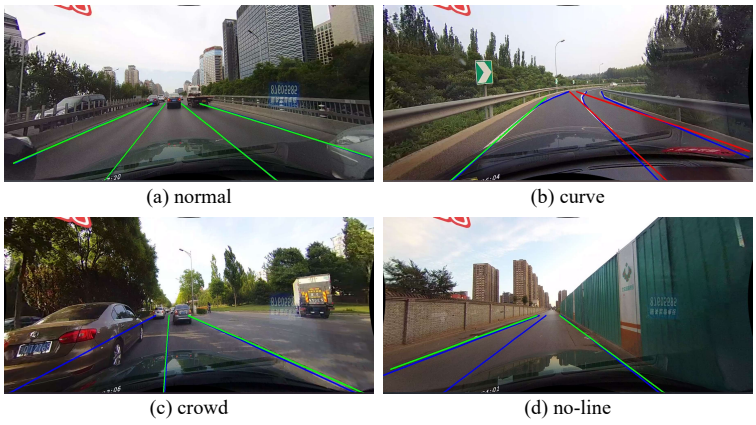


Fig. 1. Illustration of different scenarios in lane detection. Blue lines are ground truth, while green and red lines are true positive and false positive cases. (a) Lane detection is accurate in normal scenes (despite the level of features we focus on). (b) Detailed localization of lanes is inaccurate when we only utilize high-level features in the curve scene. While facing the non-visual evidence case, *e.g.*, (c) occlusions and (d) no-line, only utilizing low-level features will increase false negative samples.

Owing to the effective feature extraction capability of deep convolutional networks [3], most recent studies have focused on deep learning [4, 5] to address the task of lane detection and achieved impressive performance on mainstream datasets [5, 6]. However, there still lacks an efficient and effective lane representation. Given a front-view image captured by a camera mounted on the vehicle, segmentation-based method [7] outputs a segmentation map with per-pixel predictions and do not consider lanes as a whole unit. They predict a single feature map and overlook important semantic information because the lane itself has a slender topology and is easily occluded in the actual scene; moreover, it faces the problem of high label imbalance and time-consuming. Parameter-based [8, 9] and keypoint-based [10, 11] methods significantly improved inference speed, which model the lanes as holistic curves and treat lane line detection as a lane points localization and association problem respectively. However, those methods struggle to achieve higher performance because the polynomial coefficients are difficult to learn, and also the key points that are easily occluded will considerably affect the accuracy of lane line detection. Recently proposed anchor-based methods [12] regress the offsets between predefined fixed anchor

points and lane points, then non-maximum suppression (NMS) [13] is applied to select lane lines with the highest confidence. However, the tremendous number of fixed anchors leads to inefficient calculations in NMS procedure.

To address the aforementioned issues, we propose a novel method named DILane. For efficiency, we perform learnable anchors to replace the fixed anchors. In our observations and analysis, the distribution of lane lines in the image is statistical, *i.e.*, most of lane lines start at the bottom or lower side of the image and end at the vanishing point [14]. Besides, the distance between lane lines is generally far from each other. Thus, the tremendous number of predefined fixed anchor is inefficient, the much less learnable anchors are expected to represent where lane lines are most likely to appear after the optimization.

For effectiveness, we perform dynamic head and self-attention in parallel to gather local and global information respectively. As shown in Fig. 1, the lane line has a high probability of being occluded and damaged in practical application scenarios which is different from other detection tasks [15, 16]. Only leveraging low-level features that contain texture information will lead to an increase in false negative samples. Besides, lane lines are thin and have a long geometric shape at the same time, only using high-level features that mostly carry semantic information will cause localization inaccuracy. The proposed dynamic head conditionally enhanced high-level features with low-level features for each proposed lane instance. Inspired by iFormer [17], the self-attention is performed in parallel not only provide a possible implementation for instance interaction but also flexibly model discriminative information scattered within a wide frequency range.

Our contributions are as follows: 1) we propose learnable anchors that are the statistics of potential lane location in the image. 2) we develop a dynamic head to conditionally enhance high-level features with corresponding low-level features. 3) self-attention is utilized in parallel for global information aggregation. 4) the proposed method achieves state-of-the-art performance on two mainstream benchmarks of lane detection with high speed.

2 Related works

According to the representation of lanes, current convolutional neural network (CNN)-based lane detection methods can be divided into four main categories: segmentation-based, anchor-based, keypoint-based, and parameter-based methods, as shown in Fig. 2.

2.1 Segmentation-based methods

Segmentation-based algorithms typically adopt a pixel-wise prediction formulation; *i.e.*, they treat lane detection as a semantic segmentation task, with each pixel classified as either a lane area or background. LaneNet [18] considered lane detection as an instance segmentation problem, a binary segmentation branch and embedding branch were included to disentangle the segmented results into

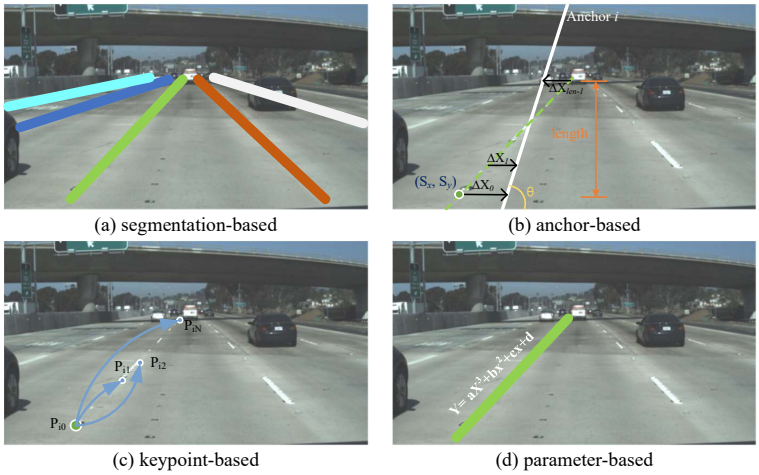


Fig. 2. Lane detection strategies. (a) Segmentation-based methods use per-pixel classification into several categories. (b) Anchor-based methods regress offsets between sampled points and predefined anchor points. (c) Keypoint-based methods use keypoint estimation and association. (d) Parameter-based methods regress parameters of lane curves.

lane instances. To distinguish different lane lines, SCNN [5] proposed a message-passing mechanism to address the no visual evidence problem, which captures the strong spatial relationship for lanes. Based on the SCNN, RESA [7] aggregates spatial information by shifting the sliced feature map recurrently in the vertical and horizontal directions. To achieve real-time requirements in practice, ENet-SAD [19] presented a knowledge distillation approach for transferring knowledge from large networks to small networks. CurveLane-NAS [20] proposed a neural architecture search (NAS) to find a better network for lane detection, which is extremely computationally expensive and requires 5,000 GPU hours per dataset.

2.2 Anchor-based methods

The methods of constructing an anchor can be divided into two types: line anchors and row anchors. In line anchor-based methods, a line-CNN [21] obtains a feature vector from each boundary position for regression and classification. LaneATT [12] predefined a set of fixed anchors for feature pooling, and proposed a attention mechanism for lane detection, which is potentially useful in other domains where the objects being detected are correlated. SGNet [22] proposed a vanishing point guided anchoring mechanism and multilevel structural constraints to improve performance. CLRNet [23] detected lanes with high-level semantic features, and then performed refinement based on low-level features. The row anchor-based approach selects the locations of lanes at predefined rows of the image, UFLD [24, 25] proposed a simple formulation of lane detection

aiming at extremely fast speeds and solving the non-visual cue problem. CondLaneNet [26] aimed to resolve lane instance-level discrimination based on conditional convolution and row anchor-based formulation.

2.3 Keypoint-based methods

Inspired by human pose estimation [27], some studies have treated lane detection as a keypoint estimation and association problem. PINet [28] used a stacked hourglass network to predict keypoint positions and cast a clustering problem of the predicted keypoints as an instance segmentation problem. In FastDraw [4], the author proposes a novel learning-based approach to decode the lane structures, which avoids the need for clustering post-processing steps. FOLOLane [10] decomposed lane detection into the subtasks of modeling local geometry and predicting global structures in a bottom-up manner. GANet [11] proposed a novel global association network to formulate lane detection from a new keypoint-based perspective that directly regresses each keypoint to its lane.

2.4 Parameter-based methods

The parameter-based methods treat lane lines as a parameter regression problem. PolyLaneNet [29] outputted polynomials that represent each lane marking in the image along with the domains for these polynomials and confidence scores for each lane. LSTR [9] developed a transformer-based network to capture long and thin structures for lanes and global context. BezierLaneNet [8] exploited the classic cubic Bezier curve because of its easy computation, stability, and high degree of freedom of transformation to model thin and long geometric shape properties of lane lines. Eigenlanes [30] propose a algorithm to detect structurally diverse road lanes in the eigenlane space which are data-driven lane descriptors, each lane is represented by a linear combination of eigenlanes.

3 Proposed method

In this paper, we propose a novel learnable anchor-based method called DILane to detect structurally diverse road lanes. Fig. 3 presents an overview of the proposed method.

The proposed DILane receives an input RGB image $I \in \mathbb{R}^{3\times H_i\times W_i}$ which is captured from a front-facing camera mounted on the vehicle, and predicts lanes $L = \{l_1, l_2, \ldots, l_N\}$. The $l_i = \{cls_i, S_{ix}, S_{iy}, \theta_i, len_i, (\Delta x_i^0, \Delta x_i^1, \ldots, \Delta x_i^{K-1})\}$, where the cls_i is classification outputs, the x- and y-coordinates of start point are defined as S_{ix} and S_{iy}, θ_i is the slope of the anchor, len_i is valid length of the lane, and K is the predefined maximum length of the lane with equally spaced coordinates in the y-axis, and $\Delta x_i \in \mathbb{R}^K$ is the horizontal offsets between the predictions and anchor lines. To generate these outputs, a backbone with FPN [31] is used to produce multi-level feature maps. The dynamic head and self-attention module receive multi-level features which are pooled by the learnable

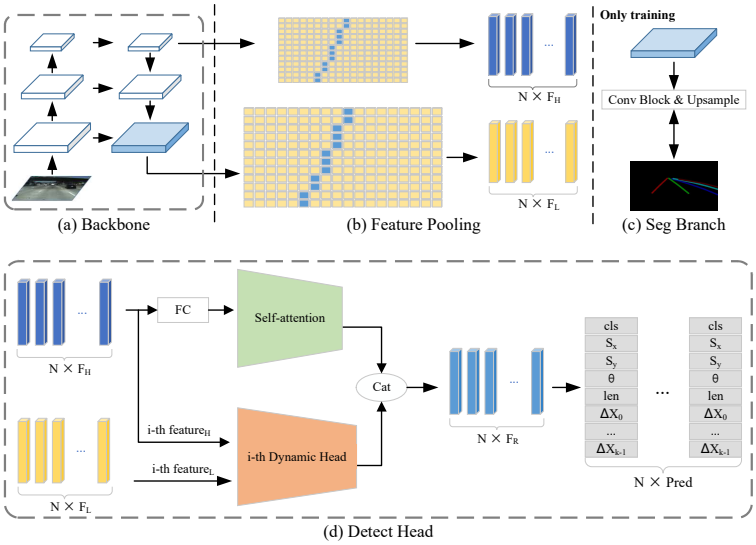


Fig. 3. Overview of DILane. (a) Backbone with FPN generates multi-level feature maps. (b) Each learnable anchor is used to pool high-level and low-level features from top and bottom layers, respectively. (c) Segmentation branch is only valid during training. (d) Detect head contains two components, the self-attention is used to aggregate global information, and dynamic head is used for local feature enhancement.

anchors from feature maps. The final feature for regression is concatenated by the outputs of dynamic head and self-attention module.

3.1 Learnable anchor

Previous anchor-based methods predefined one anchor at every possible location. Obviously, the location of lanes is statistically distributed, thus we replace a large number of fixed anchors with a small set of learnable anchors. In common object detection, objects are represented by rectangular boxes, but a lane is thin and long with strong shape priors. Thus, we define each anchor by a 4-dimensional vector $reg_i = \{S_{ix}, S_{iy}, \theta_i, len_i\}$, which denotes the normalized x and y coordinates of the starting point, direction θ and length. For every fixed $y_i^k = k \cdot \frac{H_i}{K-1}$, each predicted x-coordinate $\hat{x}_i^k$ can be calculated as following:

$$\hat{x}_i^k = \frac{1}{\tan\theta_i} \cdot (y_i^k - S_{iy}) + S_{ix} + \Delta x_i^k. \tag{1}$$

The parameters of the learnable anchor reg_i were updated with the backpropagation of the algorithm during training. Following Line-CNN [21], we simply initialize the learnable anchor to make the algorithm converge faster, as shown in Fig. 4.

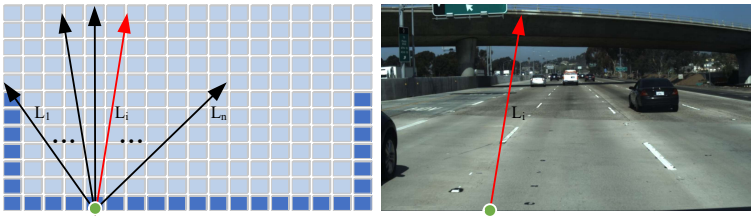


Fig. 4. Learnable anchor initialization. Anchors emitting from the starting points on the left/right/lower boundary. A straight anchor is set with a certain orientation and each starting point is associated with a group of anchors.

Conceptually, these learned anchors are the statistics of potential lane line locations in the training set and can be seen as an initial guess of the regions that are most likely to encompass the lane lines in the image, regardless of the input. Notably, proposals from a large number of fixed anchors are time-consuming and ineffective. Instead, a reasonable statistic can already qualify as a candidate. From this perspective, DILane can be categorized as a sparse lane detector.

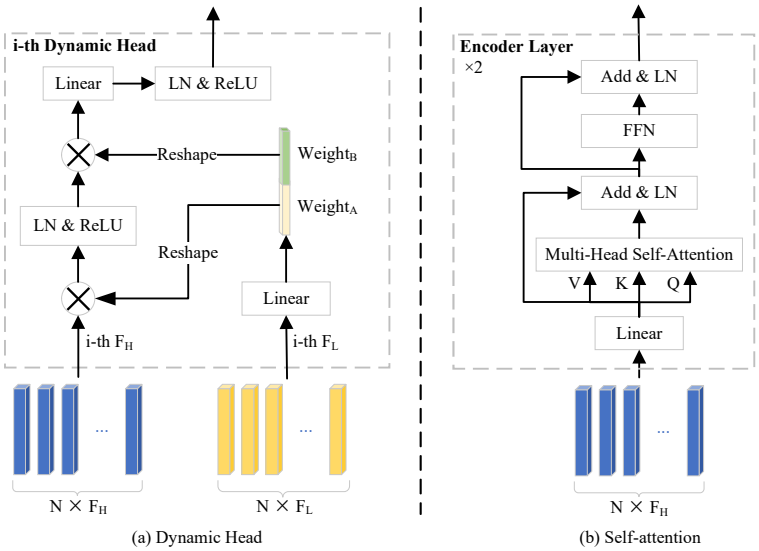


Fig. 5. Details of dynamic head and self-attention (LN: layer-norm; ⊗: 1D-convolution). (a) High-level feature and corresponding low-level feature are fed into its dynamic head to generate enhanced feature for each proposal. (b) Structure of self-attention which consists of two transformer encoder layers [32].

3.2 Dynamic head

Motivated by dynamic algorithms [33, 34], we propose the dynamic head based on conditional convolution - a convolution operation with dynamic kernel parameters [35, 36]. Sparse R-CNN [37] introduced a concept termed proposal feature $F \in \mathbb{R}^d$, which is a high-dimensional latent vector that is expected to encode rich instance characteristics. For lane detection, texture information is critical to detection accuracy when the lane line is clearly visible or less affected by the background environment; still, there are more cases where lane lines are occluded or affected by extreme lighting conditions.

Considering that bottom layers contribute more in capturing details while top layers play a significant role in modeling semantic information. To obtain more useful information dynamically under a non-visual evidence situation, we propose a dynamic head to enhance instance features, which utilizes the extracted low-level texture features to conditionally enrich high-level features.

Every anchor i has its corresponding feature vector F_H pooled from the high-level feature map and F_L pooled from the low-level feature map, which carries different semantic information. Where the part of the anchor is outside the boundaries of the feature map, both F_H and F_L are zero-padded. For each proposal instance, the high-level feature is fed into its own head, which is conditioned on a specific low-level feature. Fig. 5 illustrates the dynamic instance enhancement. in the k-th dynamic head, the k-th low-level feature F_L generates 1D-convolution kernel parameters instance-wisely for the corresponding k-th high-level feature F_H. The high-level feature $F_H \in \mathbb{R}^{H \times C}$ interact with the corresponding low-level proposal feature $F_L \in \mathbb{R}^{H \times C}$ to supplement texture information and output final enriched feature $F_R^{local} \in \mathbb{R}^C$ for next step.

3.3 Self-attention

Depending on the characteristics of the CNN to gather information from nearby pixels, each feature vector mostly carries local information. Thus, we utilize a global attention mechanism to gather the global context for each proposal feature to achieve better performance. Recent studies [38,39] have shown that the transformer has a strong capability to build long-range dependence, which achieves surprisingly high performance in many NLP tasks, *e.g.*, machine translation [40] and question answering [41]. Its success has led researchers to investigate its adaptation to the computer vision field, and Vision Transformer (ViT) [42] is a pioneer that is applied to image classification with raw image patches as input.

For each high-level feature $F_H \in \mathbb{R}^{H \times C}$, we carry out a fully connected layer to reduce the channels and generate $F'_H \in \mathbb{R}^C$. We regard each instance feature vector F'_H as a single token and put all feature vectors as a sequence into a self-attention module, which consists of two transformer encoders for gathering global information. The output of self-attention module $F_R^{global} \in \mathbb{R}^C$ concatenate with F_R^{local} which is generated by dynamic head in Sec. 3.2 and obtain $F_R = [F_R^{local}, F_R^{global}]$ for final regression.

3.4 Loss Function

Label assignment. DILane infers a fixed-size set of N predictions in a single pass through the detector, where the N is set significantly larger than the typical number of lane lines in an image. The first difficulty is scoring the predicted lanes with respect to the ground truth. Our loss produces an optimal bipartite matching between the predicted and ground truth lanes, and then optimizes the lane-specific losses. We denote the ground truth as G and the set of N predictions as $P = \{p_1, p_2, \ldots, p_N\}$. Assuming N is larger than the maximum number of lanes in the single image, we consider $G = \{g_1, g_2, \ldots, g_N\}$ as a set of N padded with $\varnothing$ (no lane). To find a bipartite matching between these two sets, we search for a permutation of N elements with the lowest cost, which is expressed as:

$$\hat{\sigma} = \underset{\sigma}{arg\,min} \sum_{i}^{N} \mathcal{L}_{match}(p_i, g_{\sigma(i)}), \tag{2}$$

where $L_{match}(p_i, g_{\sigma(i)})$ is the pairwise matching cost between a prediction p_i and ground truth with index $\sigma(i)$. This optimal assignment is computed efficiently using the Hungarian algorithm described in [43].

The matching cost considers both the class prediction and similarity of the predicted and ground truth lanes. Each element i of the ground truth set can be seen as $g_i = \{cls_i, reg_i = \{S_{ix}, S_{iy}, \theta_i, len_i\}, \Delta X_i = \{\Delta x_i^0, \Delta x_i^1, \ldots, \Delta x_i^{K-1}\}\}$. For the prediction with index $\sigma(i)$, we define the probability of class cls_i as $\hat{p}_{\sigma(i)}(cls_i)$ and the predicted parameters as $\hat{reg}_{\sigma(i)}$. With these notations, we define $L_{match}(p_i, g_{\sigma(i)})$ as:

$$\mathcal{L}_{match}(p_i, g_{\sigma(i)}) = \left(\hat{p}_{\sigma(i)}(cls_i)\right)^{\alpha} \cdot \left(1 - L_1(reg_i, \hat{reg}_{\sigma(i)}) - \mathcal{L}_{\Delta X_i}\right)^{(1-\alpha)}, \tag{3}$$

where α is set to 0.2 by default, and the above equations can be efficiently solved using the Hungarian algorithm. In Eq. 3, $\mathcal{L}_{\Delta X_i}$ is used to measure the average L1-distance between the predicted x-direction offsets and ground truth, which is defined as:

$$\mathcal{L}_{\Delta X_i} = \frac{1}{K} \sum_{j=0}^{K} L_1(\Delta x_i^j, \hat{\Delta x}_i^j). \tag{4}$$

Training loss. Our training loss includes three parts, *i.e.*, classification, matched sample, and segmentation loss, which is only used during training. The overall loss is the weighted sum of all losses:

$$\mathcal{L}_{Total} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{seg} \cdot \mathcal{L}_{seg} + \sum_{i=1}^{N} \mathbb{1}_{\{cls_i \neq \varnothing\}} \mathcal{L}_{match}(p_i, g_{\hat{\sigma}(i)}). \tag{5}$$

In Eq. 5, the $\mathcal{L}_{match}(p_i, g_{\hat{\sigma}(i)})$ is used to measure the distance between predicted lane p_i with corresponding ground truth with index $\hat{\sigma}(i)$, and is defined as:

$$\mathcal{L}_{match}(p_i, g_{\hat{\sigma}(i)}) = \lambda_{reg} \cdot \mathcal{L}_{reg}(p_i, g_{\hat{\sigma}(i)}) + \lambda_{IoU_i} \cdot \mathcal{L}_{IoU}(p_i, g_{\hat{\sigma}(i)}). \tag{6}$$

We used focal loss [44] $\mathcal{L}_{cls}$ to solve the imbalance between positive and negative examples. The regression loss $\mathcal{L}_{reg}$ is the smooth-$l1$ distance for positive lane parameters. For the x-direction offset loss $\mathcal{L}_{IoU}$, we followed the process in [23] to calculate the distance between the predicted positive samples and ground truth. Additionally, we added an auxiliary binary segmentation branch to segment the lane line or background during training, and expected the bottom-up detail perception to enforce the learning of spatial details.

4 Experiment

4.1 Datasets

To evaluate the performance of the proposed method, we conducted experiments on two well-known datasets: CULane [5] and TuSimple [6]. The CULane dataset contains 88,880 training images and $34,680$ test images, including 9 challenging scenarios collected on urban roads with 590×1640 pixels. The TuSimple dataset was collected only on highways with high-quality images, consisting of 3,626 images for training and $2,782$ images for testing, all of which had 720×1280 pixels.

4.2 Implementation details

Except when explicitly indicated, all input images were resized to $H_i \times W_i = 320 \times 800$ pixels. For all training sessions, the AdamW optimizer was used for 20 and 70 epochs on CULane and TuSimple with an initial learning rate of 5e-3. The backbone parameters were initialized by the pretrained ResNet-18/34. Data augmentation was applied to the training phase, and random affine transformation was performed (with translation, rotation, and scaling) along with random horizontal flip. Moreover, we empirically and experimentally set the number of points $K = 72$, and the intersection over union (IoU) threshold of the NMS was set to 0.5. All experiments were performed on a machine with an Intel i7-10700K processor and a single RTX 2080Ti processor.

4.3 Metrics

The F1 score is the only metric for the CULane dataset, which is based on the IoU. Because the IoU relies on areas instead of points, a lane is represented as a thick line connecting the respective points. In particular, the official metric of the dataset considers the lanes as 30-pixels-thick lines. Only when the IoU between the prediction and ground truth is greater than 0.5, can the predicted lane lines be considered as positive. The F1 score is the harmonic mean of the precision and recall, which is defined as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{7}$$

where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$.

For TuSimple dataset, the three standard official metrics are the false discovery rate $FDR = \frac{FP}{TP+FP}$, false negative rate $FNR = \frac{FN}{TP+FN}$, and the accuracy is defined as:

$$Acc = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}}, \tag{8}$$

where S_{clip} is the total number of points in the clip, and C_{clip} is the predicted lane points in the same clip. When the predicted lane point is within 20 pixels of the ground truth, it is regarded as a true positive. If a lane is to be considered as a true positive, 85% of the points must be correct.

4.4 Result

The results of our method on the CULane dataset compared with those of other popular methods are shown in Table 1. We can see that the lanes can be detected with accurate location and precise shape, even in complex scenarios. As demonstrated, our method achieves the state-of-the-art performance on the CULane benchmark with an F1 score of 79.43%. We can clearly see that the proposed method consistently outperforms other popular methods in most categories while maintaining 148+ FPS. After comparing our ResNet18-based method with other methods, the proposed method is faster than most other methods with 179 FPS and achieves an F1 score of 78.96%. These observations demonstrate the efficiency and robustness of our proposed method.

Table 1. Comparison with state-of-the-art methods on CULane dataset. F1 score ("%" is omitted) is used to evaluate the results of total and 9 sub-categories with IoU threshold equal to 0.5. For "Cross", only FP values are shown.

Method	Total	Normal	Crowd	Dazzle	Shadow	No-line	Arrow	Curve	Cross	Night	FPS
SCNN [5]	71.60	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10	7.5
RESA-R34 [7]	74.50	91.90	72.40	66.50	72.00	46.30	88.10	68.60	1896	69.80	45.5
UFLD-R18 [24]	68.40	87.70	66.00	58.40	62.80	40.20	81.00	57.90	1743	62.10	**282**
UFLD-R34 [24]	72.30	90.70	70.20	59.50	69.30	44.40	85.70	69.50	2037	66.70	170
PINet [28]	74.40	90.30	72.30	66.30	68.40	49.80	83.70	65.20	1427	67.70	25
LaneATT-R18 [12]	75.09	91.11	72.96	65.72	70.91	48.35	85.47	63.37	1170	68.95	250
LaneATT-R34 [12]	76.68	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72	171
SGNet-R18 [22]	76.12	91.42	74.05	66.89	72.17	50.16	87.13	67.02	1164	70.67	117
SGNet-R34 [22]	77.27	92.07	75.41	67.75	74.31	50.90	87.97	69.65	1373	72.69	92
Laneformer-R18 [45]	71.71	88.60	69.02	64.07	65.02	45.00	81.55	60.46	**25**	64.76	-
Laneformer-R34 [45]	74.70	90.74	72.31	69.12	71.57	47.37	85.07	65.90	26	67.77	-
[8]-R18	73.67	90.22	71.55	62.49	70.91	45.30	84.09	58.98	996	68.70	213
[8]-R34	75.57	91.59	73.20	69.20	76.74	48.05	87.16	62.45	888	69.90	150
Eigenlanes-R18 [30]	76.50	91.50	74.80	69.70	72.30	51.10	87.70	62.00	1507	71.40	-
Eigenlanes-R34 [30]	77.20	91.70	76.00	69.80	74.10	52.20	87.70	62.90	1507	71.80	-
Ours-R18	78.96	93.27	77.28	**71.98**	77.63	53.27	**89.96**	68.45	1372	74.56	179
Ours-R34	**79.43**	**93.81**	**77.70**	71.80	**78.99**	**54.12**	89.69	**70.11**	1230	**74.79**	148

Additionally, the comparison of our method on TuSimple dataset is shown in Table 2. Our method achieves the highest F1 score of 97.80% on TuSimple

dataset. Comparing with the baseline LaneATT [12], our ResNet18 version surpasses 0.90% on F1 and 1.18% on accuracy respectively. Notably, our method performs best with 2.35% on FDR rate and achieves remarkable 2.05% on FNR rate.

Table 2. Comparison with state-of-the-art methods on TuSimple dataset. All measures were computed using the official source code [5].

Method	F1(%)	Acc(%)	FDR(%)	FNR(%)
SCNN [5]	95.97	96.53	6.17	**1.80**
RESA-R34 [7]	96.93	**96.82**	3.63	2.48
UFLD-R18 [24]	87.87	95.82	19.05	3.92
UFLD-R34 [24]	88.02	95.86	18.91	3.75
LaneATT-R18 [12]	96.71	95.57	3.56	3.01
LaneATT-R34 [12]	96.77	95.63	3.53	2.92
Laneformer-R18 [45]	96.63	96.54	4.35	2.36
Laneformer-R34 [45]	95.61	96.56	5.39	3.37
BezierLaneNet-R18 [8]	95.05	95.41	5.30	4.60
BezierLaneNet-R34 [8]	95.50	95.65	5.10	3.90
Eigenlanes [30]	96.40	95.62	3.20	3.99
Ours-R18	<u>97.61</u>	96.75	<u>2.56</u>	2.22
Ours-R34	**97.80**	<u>96.82</u>	**2.35**	<u>2.05</u>

4.5 Ablation analysis

We verify the impact of the major modules through experiments on CULane dataset to show the performance and analyze each part of the proposed method. The results of the overall ablation study is presented in Table 3.

Table 3. Ablation studies on each component.

Baseline	Learnable Anchor	Dynamic Head	Att-cascade	Att-parallel	F1(%)
√	-	-	-	-	75.09
√	√	-	-	-	78.26
√	√	√	-	-	78.47
√	√	√	√	-	78.71
√	√	√	-	√	**78.96**

We take LaneATT [12] with Resnet-18 as our baseline, and gradually add the learnable anchor, dynamic head and self-attention. These three major components improve the F1 score by 3.17%, 0.21% and 0.24% respectively. The last

row of the table indicates that constructing dynamic head and self-attention in parallel brings in a gain of 0.25% F1 comparing with cascade mode.

Learnable Anchor. The number of learnable anchors has a significant impact on the detection results. We explored the impact by controlling the number of learnable anchors, as shown in Table 4.

Table 4. Effectiveness of the number of learnable anchors.

Anchors	100	200	300	500	1000
F1(%)	77.82	78.96	78.98	79.01	**79.02**
Recall(%)	73.11	73.67	73.78	73.79	**73.82**
FPS	**190**	179	160	122	96

As the proposal number increases, F1 and recall rate continue to improve, and the computational consumption also increases. In some cases, being efficient is crucial for lane detection, it might even be necessary to trade some accuracy to achieve the application's requirement. We set the number of anchors to 200 to balance efficiency and effectiveness in later experiments.

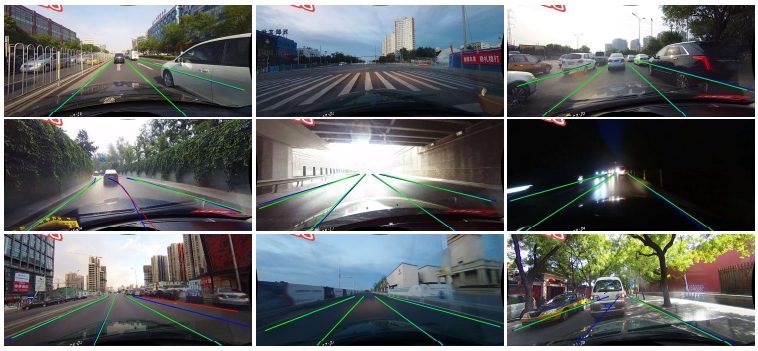
Feature enhancement. We conducted comparative experiments using different level features. We first use features from the top and bottom layers to perform feature enhancement with learnable embedding, then combined the features of different layers for comparison. The experiment are summarized in Table 5.

Table 5. Different feature enhancement settings.

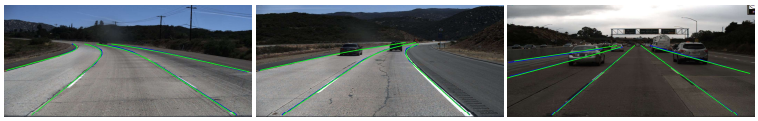
Setting	F_H	F_L	$F_L + F_H$	$F_H + F_L$
F1(%)	77.68	78.26	78.64	**78.96**

In Table 5, the "F_H" and "F_L" mean enhancing high-level and low-level features with learnable embeddings. The "$F_H + F_L$" and "$F_L + F_H$" mean enhancing high-level and low-level features with another level feature. the experiment shows enhancing high-level feature with low-level feature performs best.

Visualization. The qualitative results for CULane and TuSimple datasets are shown in Fig. 6. Particularly, we select one image from each of the nine subcategories of CULane dataset for visualization. These visualizations demonstrate that our proposed DILane is able to provide high quality lane representation.



(a) Visualization on CULane dataset



(b) Visualization on TuSimple dataset

Fig. 6. Visualization on the Tusimple and the CULane dataset. Blue lines are ground-truth, while green and red lines are true-positive and false-positive.

5 Conclusion

In this paper, we propose the Dynamic Instance-Aware Network (DILane) for lane detection. We introduce the learnable anchors which significantly improved the efficiency. Comparing with previous anchor-based methods [12, 22] which predefined 1,000 fixed anchors, our method performs better (+2.14%) which only predefined 200 learnable anchors. Notably, our proposed dynamic head and self-attention in parallel, which focus on cross-level feature enhancement and instance interaction, improved the effectiveness remarkably (+0.7%). Our method achieves the state-of-the-art performance on both CULane and TuSimple datasets with F1 score of 79.43% and 97.80% while running at 148+ FPS.

Acknowledgements This work was supported in part by the National Key R&D Program of China (2018AAA0102801 and 2018AAA0102803), and in part of the National Natural Science Foundation of China (61772424, 61702418, and 61602383).

References

1. Badue, C., Guidolini, R., Carneiro, R.V., Azevedo, P., Cardoso, V.B., Forechi, A., Jesus, L., Berriel, R., Paixao, T.M., Mutz, F., et al.: Self-driving cars: A survey. Expert Systems with Applications **165** (2021) 113816
2. Berriel, R.F., de Aguiar, E., De Souza, A.F., Oliveira-Santos, T.: Ego-lane analysis system (elas): Dataset and algorithms. Image and Vision Computing **68** (2017) 64–75
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
4. Philion, J.: 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. (2019) 11582–11591
5. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2018)
6. TuSimple: Tusimple benchmark. https://github.com/TuSimple/tusimple-benchmark/ (2020)
7. Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., Cai, D.: Resa: Recurrent feature-shift aggregator for lane detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. (2021) 3547–3554
8. Feng, Z., Guo, S., Tan, X., Xu, K., Wang, M., Ma, L.: Rethinking efficient lane detection via curve modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 17062–17070
9. Liu, R., Yuan, Z., Liu, T., Xiong, Z.: End-to-end lane shape prediction with transformers. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. (2021) 3694–3702
10. Qu, Z., Jin, H., Zhou, Y., Yang, Z., Zhang, W.: Focus on local: Detecting lane marker from bottom up via key point. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 14122–14130
11. Wang, J., Ma, Y., Huang, S., Hui, T., Wang, F., Qian, C., Zhang, T.: A keypoint-based global association network for lane detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 1392–1401
12. Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: 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. (2021) 294–302
13. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. (2017) 5561–5569
14. Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., So Kweon, I.: Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE international conference on computer vision. (2017) 1947–1955
15. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)
16. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

17. Si, C., Yu, W., Zhou, P., Zhou, Y., Wang, X., Yan, S.: Inception transformer. arXiv preprint arXiv:2205.12956 (2022)
18. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE intelligent vehicles symposium (IV), IEEE (2018) 286–291
19. Hou, Y., Ma, Z., Liu, C., Loy, C.C.: Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. (2019) 1013–1021
20. Xu, H., Wang, S., Cai, X., Zhang, W., Liang, X., Li, Z.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: European Conference on Computer Vision, Springer (2020) 689–704
21. Li, X., Li, J., Hu, X., Yang, J.: Line-cnn: End-to-end traffic line detection with line proposal unit. IEEE Transactions on Intelligent Transportation Systems **21** (2019) 248–258
22. Su, J., Chen, C., Zhang, K., Luo, J., Wei, X., Wei, X.: Structure guided lane detection. arXiv preprint arXiv:2105.05403 (2021)
23. Zheng, T., Huang, Y., Liu, Y., Tang, W., Yang, Z., Cai, D., He, X.: Clrnet: Cross layer refinement network for lane detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 898–907
24. Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection. In: European Conference on Computer Vision, Springer (2020) 276–291
25. Qin, Z., Zhang, P., Li, X.: Ultra fast deep lane detection with hybrid anchor driven ordinal classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
26. Liu, L., Chen, X., Zhu, S., Tan, P.: Condlanenet: a top-to-down lane detection framework based on conditional convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 3773–3782
27. Zhang, F., Zhu, X., Wang, C.: Single person pose estimation: A survey. arXiv preprint arXiv:2109.10056 (2021)
28. Ko, Y., Lee, Y., Azam, S., Munir, F., Jeon, M., Pedrycz, W.: Key points estimation and point instance segmentation approach for lane detection. IEEE Transactions on Intelligent Transportation Systems (2021)
29. Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Polylanenet: Lane estimation via deep polynomial regression. In: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE (2021) 6150–6156
30. Jin, D., Park, W., Jeong, S.G., Kwon, H., Kim, C.S.: Eigenlanes: Data-driven lane descriptors for structurally diverse lanes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 17163–17171
31. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 2117–2125
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: European conference on computer vision. (2020)
34. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. Advances in Neural information processing systems (2020)
35. Jia, X., De Brabandere, B., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. Advances in neural information processing systems (2016)

36. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for efficient inference. Advances in Neural Information Processing Systems **32** (2019)
37. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. (2021) 14454–14463
38. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7794–7803
39. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision, Springer (2020) 213–229
40. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33** (2020) 1877–1901
41. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
42. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
43. Stewart, R., Andriluka, M., Ng, A.Y.: End-to-end people detection in crowded scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2325–2333
44. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. (2017) 2980–2988
45. Han, J., Deng, X., Cai, X., Yang, Z., Xu, H., Xu, C., Liang, X.: Laneformer: Object-aware row-column transformers for lane detection. arXiv preprint arXiv:2203.09830 (2022)