

# Structure Guided Lane Detection

Jinming Su\*, Chao Chen, Ke Zhang, Junfeng Luo\*, Xiaoming Wei and Xiaolin Wei  
Meituan

{sujinming, chenchao60, zhangke21, luojunfeng, weixiaoming, weixiaolin02}@meituan.com

## Abstract

Recently, lane detection has made great progress with the rapid development of deep neural networks and autonomous driving. However, there exist three mainly problems including characterizing lanes, modeling the structural relationship between scenes and lanes, and supporting more attributes (*e.g.*, instance and type) of lanes. In this paper, we propose a novel structure guided framework to solve these problems simultaneously. In the framework, we first introduce a new lane representation to characterize each instance. Then a top-down vanishing point guided anchoring mechanism is proposed to produce intensive anchors, which efficiently capture various lanes. Next, multi-level structural constraints are used to improve the perception of lanes. In the process, pixel-level perception with binary segmentation is introduced to promote features around anchors and restore lane details from bottom up, a lane-level relation is put forward to model structures (*i.e.*, parallel) around lanes, and an image-level attention is used to adaptively attend different regions of the image from the perspective of scenes. With the help of structural guidance, anchors are effectively classified and regressed to obtain precise locations and shapes. Extensive experiments on public benchmark datasets show that the proposed approach outperforms state-of-the-art methods with 117 FPS on a single GPU.

## 1 Introduction

Lane detection, which aims to detect lanes in road scenes, is a fundamental perception task and has a wide range of applications (*e.g.*, ADAS [Butakov and Ioannou, 2014], autonomous driving [Chen and Huang, 2017] and high-definition map production [Homayounfar *et al.*, 2019]). Over the past years, lane detection has made significant progress and it is also used as an important element for tasks of road scene understanding, such as driving area detection [Yu *et al.*, 2020].

To address the task of lane detection, lots of learning-based methods [Pan *et al.*, 2018; Qin *et al.*, 2020] have been

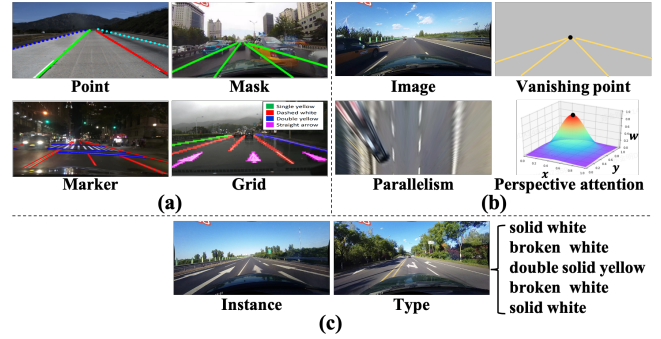


Figure 1: Challenges of lane detection. (a) Various representation. There exist many kinds of annotations [TuSimple, 2017; Pan *et al.*, 2018; Yu *et al.*, 2020; Lee *et al.*, 2017], which makes it difficult to characterize lanes in a unified way. (b) Underresearched scene structures. Lane location are strongly dependent on structural information, such as vanishing point (black point), parallelism in bird’s eye view and distance attention caused by perspective. (c) More attributes to support. Lanes have more attributes such as instance and type, which should be predicted.

proposed in recent years, achieving impressive performance on existing benchmarks [TuSimple, 2017; Pan *et al.*, 2018]. However, there still exist several challenges that hinder the development of lane detection. First, there lacks a unified and effective lane representation. As shown in (a) of Fig. 1, there exist various definitions including point [TuSimple, 2017], mask [Pan *et al.*, 2018], marker [Yu *et al.*, 2020] and grid [Lee *et al.*, 2017], which are quite different in form for different scenarios. Second, it is difficult to model the structural relationship between scenes and lanes. As displayed in (b) of Fig. 1, the structural information depending on scenes, such as location of vanishing points and parallelism of lanes, is very useful, but there is no scheme to describe it. Last, while predicting lanes, it is also important to predict other attributes including instance and type (see (c) of Fig. 1), but it is not easy to extend these for existing methods. These three difficulties are especially difficult to deal with and greatly slow down the development of lane detection. Due to these difficulties, lane detection remains a challenging vision task.

To deal with the first difficulty, many methods characterize lanes with simple fitted curves or masks. For examples, SCNN [Pan *et al.*, 2018] treats the problem as a semantic

\* Co-corresponding author.

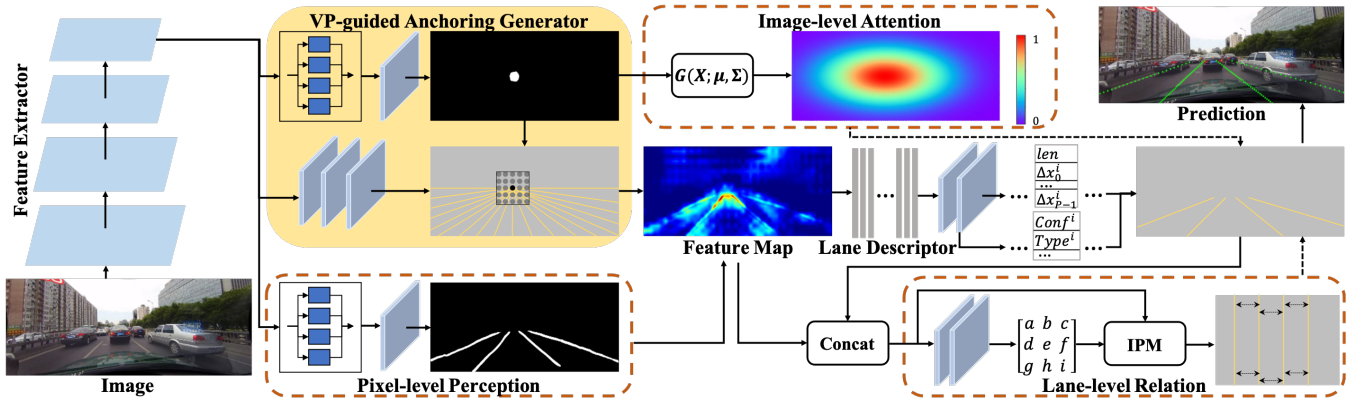


Figure 2: Framework of our approach. We first extract the common features by the extractor, which provides features for vanishing point guided anchoring and pixel-level perception. The anchoring produces intensive anchors and perception utilizes binary segmentation to promote features around lanes. Promoted features are used to classify and regress anchors with the aid of lane-level relation and image-level attention. The dashed arrow indicates the supervision, and the supervision of vanishing point and lane segmentation is omitted in the figure.

segmentation task, and introduces slice-by-slice convolutions within feature maps, thus enabling message passing. For these methods, lanes are characterized as a special form (*e.g.*, point, curve or mask), so it is difficult to support the format of marker or grid that usually has an uncertain number. Similarly, those who support the latter [Lee *et al.*, 2017] do not support the former well. To address the second problem, some methods use vanishing point or parallel relation as auxiliary information. For example, a vanishing point prediction task [Lee *et al.*, 2017] is utilized to implicitly embed a geometric context recognition capability. In these methods, they usually only pay attention to a certain kind of structural information or do not directly use it end-to-end, which leads to the structures not fully functioning and the algorithm complicated. For the last problem, some clustering- or detection-based methods are used to distinguish or classify instances. Line-CNN [Li *et al.*, 2019] utilizes line proposals as references to locate traffic curves, which forces the method to learn the feature of lanes. To these methods, they can distinguish instances and even extend to more attributes, but they usually need extra computation and have many manually designed super-parameters, which leads to poor scalability.

Inspired by these observations and analysis, we propose a novel structure guided framework for lane detection, as shown in Fig. 2. In order to characterize lanes, we propose a box-line based proposal method. In this method, the minimum circumscribed rectangle of the lane is used to distinguish instance, and its center line is used for structured positioning. For the sake of further improving lane detection by utilizing structural information, the vanishing point guided anchoring mechanism is proposed to generate intensive anchors (*i.e.*, as few and accurate anchors as possible). In this mechanism, vanishing point is learned in a segmentation manner and used to produce structural anchors top-down, which can efficiently capture various lanes. Meanwhile, we put forward multi-level structure constraints to improve the perception of lanes. In the process, the pixel-level perception is used to improve lane details with the help of lane binary segmentation, the lane-level relation aims at modeling

the parallelism properties of inter-lanes by Inverse Perspective Mapping (IPM) via a neural network, and image-level attention is to attend the image with adaptive weights from the perspective of scenes. Finally, features of lane anchors under structural guidance are extracted for accurate classification, regression and the prediction of other attributes. Experimental results on CULane and Tusimple datasets verify the effectiveness of the proposed method which achieves state-of-the-art performance and run efficiently at 117 FPS.

The main contributions of this paper include: 1) we propose a structure guided framework for lane detection, which characterize lanes and can accurately class, locate and restore the shape of unlimited lanes. 2) we introduce a vanishing point guided anchoring mechanism, in which the vanishing point is predicted and used to produce intensive anchors, which can precisely capture lanes. 3) we put forward the multi-level structural constraints, which are used to sense pixel-level unary details, model lane-level pair-wise relation and adaptively attend image-level global information.

## 2 Related Work

In this section, we review the related works that aim to resolve the challenges of lane detection in two aspects.

### 2.1 Traditional Methods

To solve the problem of lane detection, traditional methods are usually based on hand-crafted features by detecting shapes of markings and fitting the spline. [Veit *et al.*, 2008] presents a comprehensive overview of features used to detect road markings. And [Wu and Ranganathan, 2012] uses Maximally Stable Extremal Regions features and performs the template matching to detect multiple road markings. However, these approaches often fail in unfamiliar conditions.

### 2.2 Deep Learning based Methods

With the development of deep learning, methods [Pizzati and García, 2019; Van Gansbeke *et al.*, 2019; Guo *et al.*, 2020] based on deep neural networks achieve progress in lane detection. SCNN [Pan *et al.*, 2018] generalizes traditional deep

layer-by-layer convolutions to enable message passing between pixels across rows and columns. ENet-SAD [Hou *et al.*, 2019] presents a knowledge distillation approach, which allows a model to learn from itself without any additional supervision or labels. PolyLaneNet [Tabelini *et al.*, 2020] adopts a polynomial representation for the lane markings, and outputs polynomials via the deep polynomial regression. UltraFast [Qin *et al.*, 2020] treats the process of lane detection as a row-based selecting problem using global features. CurveLanes [Xu *et al.*, 2020] proposes a lane-sensitive architecture search framework to automatically capture both long-ranged coherent and accurate short-range curve information.

In these methods, different lane representations are adopted and some structural information is considered for performance improvement. However, these methods are usually based on the powerful learning ability of neural networks to learn the fitting or shapes of lanes, and the role of scene-related structural information for lanes has not been paid enough attention to and discussed.

### 3 The Proposed Approach

To address these difficulties (i.e., characterizing lanes, modeling the relationship between scenes and lanes, and supporting more attributes), we propose a novel structure guided framework for lane detection, denoted as **SGNet**. In this framework, we first introduce a new lane representation. Then a top-down vanishing point guided anchoring mechanism is proposed, and next multi-level structure constraints is used. Details of the proposed approach are described as follows.

#### 3.1 Representation

For adapting to different styles of lane annotation, we introduce a new box-line based method for lane representation. Firstly, we calculate the minimum circumscribed rectangle  $R$  (“box”) with the height  $h$  and width  $w$  for the lane instance  $L_{lane}$ . For this rectangle, center line  $L_{center}$  (“line”) perpendicular to the short side is obtained. And the angle between the positive  $X$ -axis and  $L_{center}$  in clockwise direction is  $\theta$ . In this manner,  $L_{center}$  provides the position of the lane instance, and  $h$  and  $w$  restrict the areas involved. Based on  $R$  and  $L_{center}$ , lane prediction based on points, masks, markers, grids and other formats can be performed. In this paper, the solution based on key points of lane detection is taken just because of the point-based styles of lane annotation in public datasets (e.g., CULane [TuSimple, 2017] and Tusimple [Pan *et al.*, 2018]).

Inspired by existing methods [Li *et al.*, 2019; Chen *et al.*, 2019; Qin *et al.*, 2020], we define key points of the lane instance with equally spaced  $y$  coordinates  $Y = \{y_i\}$  and  $y_i = \frac{H}{P-1} \cdot i$  ( $i = 1, 2, \dots, P-1$ ), where  $P$  means the number of all key points through image height, which is fixed on images with same height  $H$  and width  $W$ . Accordingly, the  $x$  coordinates of the lane is expressed as  $X = \{x_i\}$ . For the convenience of expression, the straight line equation of  $L_{center}$  is defined as

$$ax + by + c = 0, a \neq 0 \text{ or } b \neq 0 \quad (1)$$

where  $a$ ,  $b$  and  $c$  can be easily computed by  $\theta$  and any point on  $L_{center}$ . Next, when the  $y$  coordinate of the center line is

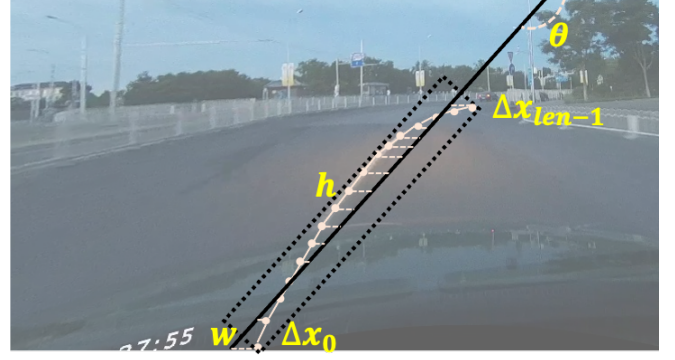


Figure 3: Lane representation.

$y_i$ , we can compute the corresponding  $x$  coordinate as

$$x_i = L_{center}(y_i) = \frac{-c - by_i}{a}, a \neq 0. \quad (2)$$

Then, we define the offset of  $x$  coordinate  $\Delta X$  between the lane  $L_{lane}$  and center line  $L_{center}$  as

$$\begin{aligned} \Delta X &= \{\Delta x_i\} = \{x_i - \frac{-c - by_i}{a}\}, \\ X &= \{\frac{-c - by_i}{a}\} + \Delta X. \end{aligned} \quad (3)$$

Therefore, based on  $L_{center}$  and  $\Delta X$ , we can calculate the lane instance  $L_{lane}$ . Usually, it is easier to learn  $L_{center}$  and  $\Delta X$  than the directly fitting key points of  $L_{lane}$ .

#### 3.2 Feature Extractor

To see Fig. 2, SGNet takes ResNet [He *et al.*, 2016] as the feature extractor, which is modified to remove the last global pooling and fully connected layers for the pixel-level prediction task. Feature extractor has five residual modules for encoding, named as  $\mathcal{E}_i(\pi_i)$  with parameters  $\pi_i$  ( $i = 1, 2, \dots, 5$ ). To obtain larger feature maps, we convolve  $\mathcal{E}_5(\pi_5)$  by a convolutional layer with 256 kernels of  $3 \times 3$  and then  $\times 2$  up-sample the features, followed by an element-wise summation with  $\mathcal{E}_4(\pi_4)$  to obtain  $\mathcal{E}'_4(\pi'_4)$ . Finally, for a  $H \times W$  input image, a  $\frac{H}{16} \times \frac{W}{16}$  feature map is output by the feature extractor.

#### 3.3 Vanishing Point Guided Anchoring

In order to learn the lane representation, there are two main ways to learn the center line  $L_{center}$  and  $x$  offset  $\Delta X$ . The first way is to learn the determined  $L_{center}$  directly with angle, number and position regression, which is usually difficult to achieve precise results because of the inherent difficulty of regression tasks. The second way is based on mature detection tasks, using dense anchors to classify, regress and then obtain proposals representing the lane instance. And the second one has been proved to work well in general object detection tasks, so we choose it as our base model.

To learn the center line  $L_{center}$  and  $x$  offset  $\Delta X$  well, we propose a novel vanishing point guided anchoring mechanism (named as VPG-Anchoring). The vanishing point (VP) provides strong characterization of geometric scene, representing the end of the road and also the “virtual” point where



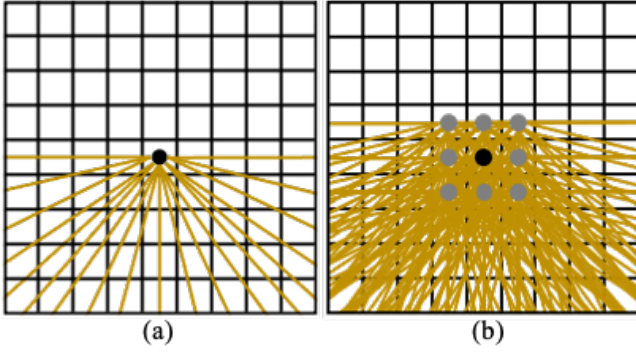


Figure 4: VP-guided anchoring mechanism. Anchors (golden lines) generated based on (a) the vanishing point (black point) and (b) the area around vanishing point (black and gray points).

the lanes intersect in the distance. Since VP is the intersection point of lanes, lanes in the scene must pass through VPs, and lines that do not pass through VPs are not lanes in the scene with high probability. Therefore, dense lines radiated from VPs can theoretically cover all lanes in the image, which is equivalent to reducing the generation space of anchors from  $\mathbb{R}^{H \times W \times N_{proposal}}$  to  $\mathbb{R}^{N_{proposal}}$ .  $N_{proposal}$  represents the number of anchors generated at one pixel.

As shown in Fig. 2, the features map  $\mathcal{E}'_4(\pi'_4)$  is feed to VPG-Anchoring. In the mechanism, VP is predicted by a simple branch, which is implemented by a multi-scale context-aware atrous spatial pyramid pooling (ASPP) [Chen *et al.*, 2018] followed by a convolutional layer with 256 kernels of  $3 \times 3$  and a softmax activation. The VP prediction branch is denoted as  $\phi_V(\pi_V)$  with parameters  $\pi_V$ .

Usually, VP is not annotated in lane datasets, such as CU-Lane [Pan *et al.*, 2018], so we average the intersection points of the center lines of all lane instances and get the approximate VP. In addition, a single point is usually difficult to predict, so we expand the area of VP to a radius of 16 pixels and use segmentation algorithm to predict. To achieve this, we expect the output of  $\phi_V(\pi_V)$  to approximate the ground-truth masks of VP (represented as  $G_V$ ) by minimizing the loss

$$\mathcal{L}_V = BCE(\phi_V(\pi_V), G_V), \quad (4)$$

where  $BCE(\cdot, \cdot)$  represents the pixel-level binary cross-entropy loss function.

In order to ensure that generated anchors are dense enough, we choose a  $W_{anchor} \times W_{anchor}$  rectangular area with VP as the center, and take one point every  $S_{anchor}$  to generate anchors. For each point, anchors are generated every  $A_{anchor}$  angle ( $A_{anchor} \in [0, 180]$ ) as shown in Fig. 4. In this way, anchors are targeted, intensive and not redundant, compared with general full-scale uniform generation and even specially designed methods for lanes [Li *et al.*, 2019]. Note that anchors run through the whole image, and only the part below VP is shown for convenient display in Figs. 2 and 4.

### 3.4 Classification and Regression

In order to classify and regress the generated anchors, we extract high-level feature maps based on  $\mathcal{E}_4(\pi_4)$  with several convolutional layers. The feature map is named as  $F_A \in$

$\mathbb{R}^{H' \times W' \times C'}$ , where  $H'$ ,  $W'$  and  $C'$  are the height, width and channel of  $F_A$ . For each anchor  $L_{lane}$ , the channel-level features of each point on anchors are extracted from  $F_A$  to obtain lane descriptor  $D_A \in \mathbb{R}^{H' \times C'}$ , which are used to classify the existence  $Conf^{L_{lane}}$  and regress  $x$  offsets  $\Delta X^{L_{lane}}$  including the length  $len$  of lanes. To learn these, we expect the output to approximate the ground-truth existence  $GConf^{L_{lane}}$  and  $x$  offsets  $G\Delta X^{L_{lane}}$  by minimizing the loss

$$\begin{aligned} \mathcal{L}_C &= \sum_{L_{lane}=0}^{L-1} BCE(Conf^{L_{lane}}, GConf^{L_{lane}}), \\ \mathcal{L}_R &= \sum_{L_{lane}=0}^{L-1} SL1(\Delta X^{L_{lane}}, G\Delta X^{L_{lane}}), \end{aligned} \quad (5)$$

where  $SL1(\cdot, \cdot)$  means smooth L1 loss and  $L$  means the number of proposals. Finally, Line-NMS [Li *et al.*, 2019] is used to obtain the finally result with confidence thresholds.

### 3.5 Multi-level Structure Constraints

In order to further improve lane perception, we ask for the structural relationship between scenes and lanes, and deeply explore the pixel-level, lane-level and image-level structures.

**Pixel-level Perception.** The top-down VPG-Anchoring mechanism covers the structures and distribution of lanes. At the same time, there is a demand of bottom-up detail perception, which ensures that lane details are restored and described more accurately. For the sake of improving the detail perception, we introduce lane segmentation branch to location lane locations and promote pixel-level unary details. As shown in Fig. 2, the lane segmentation branch has the same input and similar network structure with the VP prediction branch. The lane segmentation branch is denoted as  $\phi_P(\pi_P)$  with parameters  $\pi_P$ . To segment lanes, we expect the output of  $P_P = \phi_P(\pi_P)$  to approximate the ground-truth masks of binary lane mask (represented as  $G_P$ ) by minimizing the loss

$$\mathcal{L}_P = BCE(P_P, G_P). \quad (6)$$

To promote the pixel-level unary details, we weight the input features  $F_A$  by the following operation

$$M_A = F_A \otimes P_P + F_A, \quad (7)$$

where  $M_A$  are feed to classify and regress instead of  $F_A$ .

**Lane-level Relation.** In fact, lanes conform to certain rules in the construction process, and the most important one is that the lanes are parallel. Due to imaging reasons, this relationship is no longer maintained after perspective transformation, but it can be modeled potentially. To model the lane-level relation, we conduct IPM by the  $H$  Matrix [Neven *et al.*, 2018] via a neural network. After learning  $H$ , the lane instance  $L_{lane}$  can be transformed to  $L'_{lane}$  on bird's eye view, where different instances are parallel. Formally, we define the relationship between lanes as follows. For two lane instances  $L_{lane1}$  and  $L_{lane2}$  in the image, they are projected to the bird's-eye view through the learned  $H$  matrix, and the corresponding instance  $L'_{lane1}$  and  $L'_{lane2}$  are obtained. The two instances can be fitted to the following linear equations:

$$\begin{aligned} a_1 * x + b_1 * y + c_1 &= 0, \\ a_2 * x + b_2 * y + c_2 &= 0. \end{aligned} \quad (8)$$

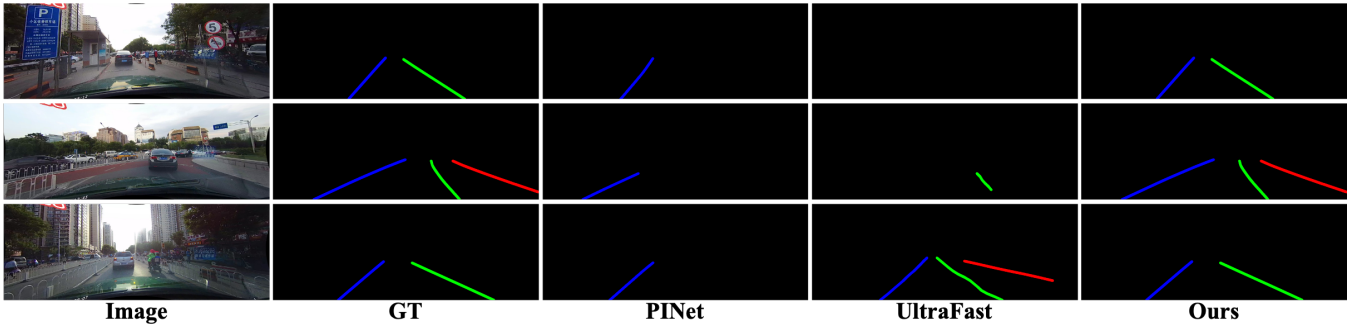


Figure 5: Qualitative comparisons of the state-of-the-art algorithms and our approach.

|                   | Total                    | Normal                   | Crowd                    | Dazzle                   | Shadow                   | No line                  | Arrow                    | Curve                    | Cross                   | Night                    | FPS                    |
|-------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------|--------------------------|------------------------|
| DeepLabV2-50      | 66.70                    | 87.40                    | 64.10                    | 54.10                    | 60.70                    | 38.10                    | 79.00                    | 59.80                    | 2505                    | 60.60                    | -                      |
| SCNN              | 71.60                    | 90.60                    | 69.70                    | 58.50                    | 66.90                    | 43.40                    | 84.10                    | 64.40                    | 1990                    | 66.10                    | 8                      |
| FD                | -                        | 85.90                    | 63.60                    | 57.00                    | 59.90                    | 40.60                    | 79.40                    | 65.20                    | 7013                    | 57.80                    | -                      |
| ENet-SAD          | 70.80                    | 90.10                    | 68.80                    | 60.20                    | 65.90                    | 41.60                    | 84.00                    | 65.70                    | 1998                    | 66.00                    | 75                     |
| PointLane         | 70.20                    | 88.00                    | 68.10                    | 61.50                    | 63.30                    | 44.00                    | 80.90                    | 65.20                    | 1640                    | 63.20                    | -                      |
| RONELD            | 72.90                    | -                        | -                        | -                        | -                        | -                        | -                        | -                        | -                       | -                        | -                      |
| PINet             | 74.40                    | 90.30                    | 72.30                    | 66.30                    | 68.40                    | <b>49.80<sub>3</sub></b> | 83.70                    | 65.60                    | <b>1427<sub>3</sub></b> | 67.70                    | 25                     |
| ERFNet-E2E        | 74.00                    | <b>91.00<sub>3</sub></b> | <b>73.10<sub>3</sub></b> | 64.50                    | <b>74.10<sub>2</sub></b> | 46.60                    | <b>85.80<sub>3</sub></b> | <b>71.90<sub>1</sub></b> | 2022                    | 67.90                    | -                      |
| IntRA-KD          | 72.40                    | -                        | -                        | -                        | -                        | -                        | -                        | -                        | -                       | -                        | 98                     |
| UltraFast-18      | 68.40                    | 87.70                    | 66.00                    | 58.40                    | 62.80                    | 40.20                    | 81.00                    | 57.90                    | 1743                    | 62.10                    | <b>323<sub>1</sub></b> |
| UltraFast-34      | 72.30                    | 90.70                    | 70.20                    | 59.50                    | 69.30                    | 44.40                    | 85.70                    | <b>69.50<sub>3</sub></b> | 2037                    | 66.70                    | <b>175<sub>2</sub></b> |
| CurveLanes        | <b>74.80<sub>3</sub></b> | 90.70                    | 72.30                    | <b>67.70<sub>2</sub></b> | 70.10                    | 49.40                    | <b>85.80<sub>3</sub></b> | 68.40                    | 1746                    | <b>68.90<sub>3</sub></b> | -                      |
| <b>Ours-Res18</b> | <b>76.12<sub>2</sub></b> | <b>91.42<sub>2</sub></b> | <b>74.05<sub>2</sub></b> | <b>66.89<sub>3</sub></b> | <b>72.17<sub>3</sub></b> | <b>50.16<sub>2</sub></b> | <b>87.13<sub>2</sub></b> | 67.02                    | <b>1164<sub>1</sub></b> | <b>70.67<sub>2</sub></b> | <b>117<sub>3</sub></b> |
| <b>Ours-Res34</b> | <b>77.27<sub>1</sub></b> | <b>92.07<sub>1</sub></b> | <b>75.41<sub>1</sub></b> | <b>67.75<sub>1</sub></b> | <b>74.31<sub>1</sub></b> | <b>50.90<sub>1</sub></b> | <b>87.97<sub>1</sub></b> | <b>69.65<sub>2</sub></b> | <b>1373<sub>2</sub></b> | <b>72.69<sub>1</sub></b> | 92                     |

 Table 1: Comparisons with state-of-the-art methods on CULane dataset. F1-measure score (“%” is omitted) is used to evaluate the results of total and 8 sub-categories. For Cross, only FP are shown. The top three results are in **red<sub>1</sub>**, **green<sub>2</sub>** and **blue<sub>3</sub>** fonts with a footnote.

In these two equations, under the condition that  $y$  is equal, the difference of  $x$  is always constant. Thus we can get that  $a_1 * b_2 = a_2 * b_1$ . Expanding to all instances, lane-level relation can be formulated as

$$L_{\mathcal{L}} = \sum_{i=0, j=0, i \neq j}^{L-1} L1(a_i b_j - a_j b_i). \quad (9)$$

**Image-level Attention.** In the process of camera imaging, distant objects are small after projection. Usually, the distant information of lanes is not prominent visually, but they are equally important. After analysis, it is found that the distance between lanes and VP reflects the inverse proportion to scales in imaging. Therefore, we generate perspective attention map PAM based on VP, which is based on the strong assumption that the attention and distance after imaging satisfies two-dimensional gaussian distribution. PAM ensures the attention of different regions by adaptively restricting the classification and regression loss (from Eq. 5) as follows.

$$L_{\mathcal{I}} = \sum_{L_{lane}=0}^{L-1} \sum_{p=0}^{P-1} L1(\Delta x_p^{L_{lane}}, G \Delta x_p^{L_{lane}}) \cdot (1 + |E(x_p^{L_{lane}}, y_p^{L_{lane}})|), \quad (10)$$

where  $|\cdot|$  means normalized to  $[0, 1]$ .

By taking the losses of Eqs.(4),(5),(6),(9) and (10), the

overall learning objective can be formulated as follows:

$$\min_{\mathbb{P}} \mathcal{L}_{\mathcal{V}} + \mathcal{L}_{\mathcal{C}} + \mathcal{L}_{\mathcal{R}} + \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{L}} + \mathcal{L}_{\mathcal{I}}, \quad (11)$$

where  $\mathbb{P}$  is the set of  $\{\{\pi_i\}_{i=1}^5, \pi'_4, \pi_{\mathcal{V}}, \pi_{\mathcal{C}}, \pi_{\mathcal{R}}, \pi_{\mathcal{P}}, \pi_{\mathcal{L}}\}$ , and  $\pi_{\mathcal{C}}, \pi_{\mathcal{R}}$  and  $\pi_{\mathcal{L}}$  are the parameters of classification, regression and lane-level relation subnetworks, respectively.

## 4 Experiments and Results

### 4.1 Experimental Setup

**Dataset.** To evaluate the performance of the proposed method, we conduct experiments on CULane [Pan *et al.*, 2018] and Tusimple [TuSimple, 2017] dataset. CULane dataset has a split with 88,880/9,675/34,680 images for train/val/test and Tusimple dataset is divided into three parts: 3,268/358/2,782 for train/val/test.

**Metrics.** For CULane, we use F1-measure score as the evaluation metric. Following [Pan *et al.*, 2018], we treat each lane as a line with 30 pixel width and compute the intersection-over-union (IoU) between groundtruths and predictions with a threshold of 0.5. For Tusimple, the official metric (Accuracy) is used as the evaluation criterion, which evaluates the correction of predicted lane points.

**Training and Inference.** We use Adam optimization algorithm to train our network end-to-end by optimizing the loss in Eq. (11). In the optimization process, the parameters of

|                   | Accuracy                 | FPS                    |
|-------------------|--------------------------|------------------------|
| DeepLabV2-18      | 92.69                    | 40                     |
| DeepLabV2-34      | 92.84                    | 20                     |
| SCNN              | <b>96.53<sub>2</sub></b> | 8                      |
| FD                | 94.90                    | -                      |
| ENet-SAD          | <b>96.64<sub>1</sub></b> | <b>75<sub>3</sub></b>  |
| Cascaded-CNN      | 95.24                    | 60                     |
| PolyLaneNet       | 93.36                    | <b>115<sub>1</sub></b> |
| <b>Ours-Res34</b> | <b>95.87<sub>3</sub></b> | <b>92<sub>2</sub></b>  |

Table 2: Comparisons with state-of-the-arts on Tusimple.

feature extractor are initialized by the pre-trained ResNet-18/34 model and “poly” learning rate policy are employed for all experiments. The training images are resized to the resolution of  $360 \times 640$  for faster training, and applied affine and flipping. And we train the model for 10 epochs on CULane and 60 epochs on TuSimple. Moreover, we empirically and experimentally set the number of points  $P = 72$ , the width of rectangular  $W_{anchor} = 40$ , anchor strides  $S_{anchor} = 5$  and anchor angle interval  $A_{anchor} = 5$ .

## 4.2 Comparisons with State-of-the-art Methods

We compare our approach with state-of-the-arts including DeeplabV2 [Chen *et al.*, 2017], SCNN [Pan *et al.*, 2018], FD [Pillion, 2019], ENet-SAD [Hou *et al.*, 2019], PointLane [Chen *et al.*, 2019], RONELD [Chng *et al.*, 2020], PINet [Ko *et al.*, 2020], ERFNet-E2E [Yoo *et al.*, 2020], IntRA-KD [Hou *et al.*, 2020], UltraFast [Qin *et al.*, 2020], CurveLanes [Xu *et al.*, 2020], Cascaded-CNN [Pizzati *et al.*, 2019] and PolyLaneNet [Tabelini *et al.*, 2020].

We compare our approach with 10 state-of-the-art methods on CULane dataset, as listed in Tab. 1. Comparing our ResNet34-based method with others, we can see that the proposed method consistently outperforms other methods across total and almost all categories. For the total dataset, our method is noticeably improved from 74.80% to 77.27% compared with the second best method. Also, it is worth noting that our method is significantly better on Crowd (+2.31%), Arrow (+2.17%) and Night (+3.79%) compared with second best methods, respectively. In addition, we also obviously lower FP on Cross by 3.78% relative to the second best one. As for Curve, we are slightly below the best method (ERFNet-E2E), which conducts special treatment for curve points while maybe damaging other categories. Moreover, our method has a faster FPS than almost all results. These observations present the efficiency and robustness of our proposed method and validate that VPG-Anchoring and multi-level structures are useful for the task of lane detection.

Some examples generated by our approach and other state-of-the-art algorithms are shown in Fig. 5. We can see that lanes can be detected with accurate location and precise shape by the proposed method, even in complex situations. These visualizations indicate that the proposed lane representation has a good characterization of lanes, and also show the superiority of the proposed method.

Moreover, we list the comparisons on Tusimple as shown in Tab. 2. It can be seen that our method is competitive in highway scenes without adjustment, which further proves the

|              | VPG-A | Pixel | Lane | Image | Total |
|--------------|-------|-------|------|-------|-------|
| Base         |       |       |      |       | 71.98 |
| Base+V-F     | ✓     |       |      |       | 74.08 |
| Base+V       | ✓     |       |      |       | 74.27 |
| Base+V+P     | ✓     | ✓     |      |       | 76.30 |
| Base+V+P+L   | ✓     | ✓     | ✓    |       | 76.70 |
| <b>SGNet</b> | ✓     | ✓     | ✓    | ✓     | 77.27 |

Table 3: Performance of different settings of the proposed method. “-A” means “Anchoring”.

effectiveness of structural information for lane detection.

## 4.3 Ablation Analysis

To validate the effectiveness of different components of the proposed method, we conduct several experiments on CULane to compare the performance variations of our methods.

**Effectiveness of VPG-Anchoring.** To investigate the effectiveness of the proposed VPG-Anchoring, we conduct ablation experiments and introduce three different models for comparisons. The first setting is only the feature extractor and the subnetwork of classification and regression, which is regarded as “Base” model. In Base, anchor is generated uniformly at all positions of the feature map, and  $A_{anchor}$  is lowered to ensure the same number with SGNet. In addition, we conduct another model (“Base+V”) by adding VPG-Anchor. And we also replace the  $L_{center}$  by straight line fitted directly by key points as the “Base+V-F” to explore the importance of VP. The comparisons of above models are listed in Tab. 3. We can observe that the VPG-Anchoring greatly improve the performance of Base model, which verifies the effectiveness of this mechanism. In addition, comparing Base+V with Base+V-F, we find the proposed approximate VP in lane presentation is better than the one by direct fitting.

**Effectiveness of Multi-level Structures.** To explore the effectiveness of the pixel-level, lane-level and image-level structures, we conduct another experiments by combining the pixel-level perception with “Base+V” as “Base+V+P” and adding lane-level relation to “Base+V+P” as “Base+V+P+L”. From the last four rows of Tab. 3, we can find that the performance of lane detection can be continuously improved by pixel-, lane- and image-level structures, which validates that the three levels of constrains are compatible with each other, and can be used together to gain performance.

## 5 Conclusion

In this paper, we rethink the difficulties that hinder the development of lane detection and propose a structure guided framework. In this framework, we introduce a new lane representation to meet the demands of various lane representations. Based on the representation, we propose a novel vanishing point guided anchoring mechanism to generate intensive anchors for efficiently capturing lanes. In addition, multi-level structure constraints is modeled to improve lane perception. Extensive experiments on benchmark datasets validates the effectiveness of the proposed approach with fast inference and shows that the perspective of modeling and utilization of structure information is useful for lane detection.

## References

- [Butakov and Ioannou, 2014] Vadim A Butakov and Petros Ioannou. Personalized driver/vehicle lane change models for adas. *IEEE TVT*, 64(10):4422–4431, 2014.
- [Chen and Huang, 2017] Zhilu Chen and Xinming Huang. End-to-end learning for lane keeping of self-driving cars. In *IEEE IV*, 2017.
- [Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.
- [Chen *et al.*, 2018] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.
- [Chen *et al.*, 2019] Zhenpeng Chen, Qianfei Liu, and Chenfan Lian. Pointlanenet: Efficient end-to-end cnns for accurate real-time lane detection. In *IEEE IV*, 2019.
- [Chng *et al.*, 2020] Zhe Ming Chng, Joseph Mun Hung Lew, and Jimmy Addison Lee. Roneld: Robust neural network output enhancement for active lane detection. *arXiv preprint arXiv:2010.09548*, 2020.
- [Guo *et al.*, 2020] Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-lanenet: A generalized and scalable approach for 3d lane detection. In *ECCV*, 2020.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [Homayounfar *et al.*, 2019] Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. Dagmapper: Learning to map by discovering lane topology. In *ICCV*, 2019.
- [Hou *et al.*, 2019] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In *ICCV*, 2019.
- [Hou *et al.*, 2020] Yuenan Hou, Zheng Ma, Chunxiao Liu, Tak-Wai Hui, and Chen Change Loy. Inter-region affinity distillation for road marking segmentation. In *CVPR*, 2020.
- [Ko *et al.*, 2020] Yeongmin Ko, Jiwon Jun, Donghwyu Ko, and Moongu Jeon. Key points estimation and point instance segmentation approach for lane detection. *arXiv preprint arXiv:2002.06604*, 2020.
- [Lee *et al.*, 2017] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *ICCV*, 2017.
- [Li *et al.*, 2019] 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.
- [Neven *et al.*, 2018] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *IEEE IV*, 2018.
- [Pan *et al.*, 2018] Xingang Pan, Jianping Shi, Ping Luo, Xiaoogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*, 2018.
- [Phillion, 2019] Jonah Phillion. Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. In *CVPR*, 2019.
- [Pizzati and García, 2019] Fabio Pizzati and Fernando García. Enhanced free space detection in multiple lanes based on single cnn with scene identification. In *IEEE IV*, 2019.
- [Pizzati *et al.*, 2019] Fabio Pizzati, Marco Allodi, Alejandro Barrera, and Fernando García. Lane detection and classification using cascaded cnns. In *International Conference on Computer Aided Systems Theory*, 2019.
- [Qin *et al.*, 2020] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *ECCV*, 2020.
- [Tabelini *et al.*, 2020] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixão, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. *arXiv preprint arXiv:2004.10924*, 2020.
- [TuSimple, 2017] TuSimple. Tusimple lane detection challenge. <http://benchmark.tusimple.ai/#/>, 2017. Accessed: 2017.
- [Van Gansbeke *et al.*, 2019] 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 *ICCV Workshops*, 2019.
- [Veit *et al.*, 2008] Thomas Veit, Jean-Philippe Tarel, Philippe Nicolle, and Pierre Charbonnier. Evaluation of road marking feature extraction. In *IEEE Conference on Intelligent Transportation Systems*, 2008.
- [Wu and Ranganathan, 2012] Tao Wu and Ananth Ranganathan. A practical system for road marking detection and recognition. In *IEEE IV*, 2012.
- [Xu *et al.*, 2020] Hang Xu, Shaoju Wang, Xinyue Cai, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In *ECCV*, 2020.
- [Yoo *et al.*, 2020] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyounghoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-end lane marker detection via row-wise classification. In *CVPR Workshops*, 2020.
- [Yu *et al.*, 2020] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.