

CLRerNet: Improving Confidence of Lane Detection with LaneIoU

Hiroto Honda Yusuke Uchida
 GO Inc.

{hiroto.honda, yusuke.uchida}@goinc.jp

Abstract

Lane marker detection is a crucial component of the autonomous driving and driver assistance systems. Modern deep lane detection methods with anchor-based lane representation exhibit excellent performance on lane detection benchmarks. Through preliminary oracle experiments, we firstly disentangle the lane representation components to determine the direction of our approach. We show that correct lane positions are already among the predictions of an existing anchor-based detector, and the confidence scores that accurately represent intersection-over-union (IoU) with ground truths are the most beneficial. Based on the finding, we propose LaneIoU that better correlates with the metric, by taking the local lane angles into consideration. We develop a novel detector coined CLRerNet featuring LaneIoU for the target assignment cost and loss functions aiming at the improved quality of confidence scores. Through careful and fair benchmark including cross validation, we demonstrate that CLRerNet outperforms the state-of-the-art by a large margin - enjoying F1 score of 81.43% compared with 80.47% of the existing method on CULane, and 86.47% compared with 86.10% on CurveLanes. Code and models are available at <https://github.com/hirotomusiker/CLRerNet>.

1. Introduction

Lane (marker) detection plays an important role in the autonomous driving and driver assistance systems. Like other computer vision tasks, emergence of convolutional neural networks (CNNs) has brought rapid progress on lane detection performance. Modern lane detection methods are grouped into five categories. Segmentation-based [18, 27] and keypoint-based [21] methods regard lanes as segmentation mask and keypoints respectively. The parametric methods [16, 25] utilize curve parameters to regress lane shapes. The row-based and anchor-based methods [15, 19, 20, 24, 28] regards a lane as a set of coordinates on the certain horizontal lines. The first two methods are the bottom-up detection paradigms, where the lane positions are directly detected

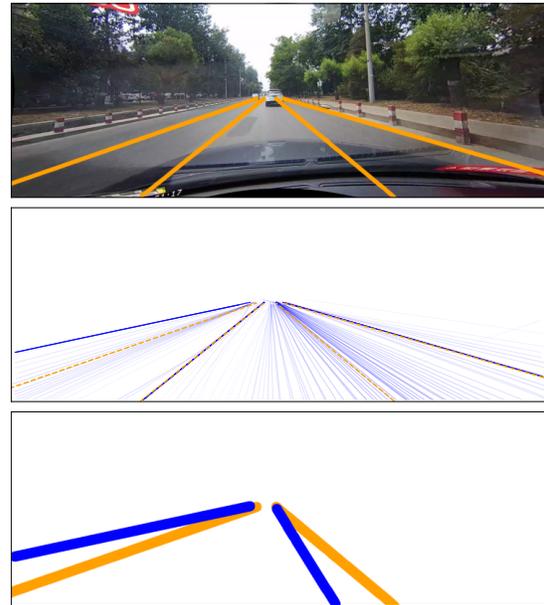


Figure 1. Lane detection example. top: ground truth. middle: all the predictions (blue, deeper for higher confidence scores) and ground truths (dashed orange). bottom: example of metric IoU calculation by comparing segmentation masks of predictions (blue) and GTs (orange). Best viewed in color.

in the image and grouped into lane instances afterwards. The latter three are the top-down instance detection methods where each lane detection is regarded as both global lane instance and a set of local lane points. The anchor-based method is the de-facto standard in terms of detection performance among the above method types. We choose the best-performing CLRerNet [28] from the anchor-based methods as our baseline.

The performance of lane detection relies on lane point localization and instance-wise classification. The lane detection benchmarks [7, 18] employ segmentation-mask-based intersection-over-union (IoU) between predicted and ground-truth (GT) lanes as an evaluation metric. The predicted lanes whose scores are above the predefined thresh-

old are treated as valid predictions to calculate F1 score. Therefore, the predicted lanes with large segmentation-based IoUs with GTs should have large classification scores.

To determine the direction of our approach, we firstly conduct the preliminary oracle experiments by replacing the confidence score, anchor parameters and length of each prediction with the oracle values. By making the confidence scores oracle, the F1 score goes to near-perfect 98.47%. The result implies the correct lanes are already among the predictions, however the confidence scores need to be predicted accurately representing the metric IoU. Fig. 1 (middle) shows the comparison between all the predictions (blue) and GTs (dashed orange). The color depth of the predictions is proportional to their confidence scores. The left-most prediction is the high-confidence false positive that misses the ground truth, however there is a correct low-confidence prediction near the GT which has a high IoU.

The next question is: how could the segmentation-based IoU be implemented as a learning target? In the anchor-based and row-based methods, the prediction and GT lanes are both represented as sets of x coordinates at the fixed rows. [28] introduces the LineIoU loss to measure the intersection and union row by row and sum them up respectively. However, this approach is not equivalent to the segmentation-based IoU especially for the non-vertical, tilted lanes (e.g. Fig. 1 bottom) or curves. We introduce the novel IoU coined *LaneIoU*, which takes local lane angles of the lanes into account. The LaneIoU integrates the angle-aware intersection and union of each row to match the segmentation-based IoU.

The anchor-based methods learn global lane probability scores for each anchor. The dynamic sample assignment employed in the recent object detector [4, 5] is also effective for lane detection training [28]. The IoU matrix and the cost matrix between the predicted lanes and the GTs respectively determine the number of anchors to assign for each GT and which anchors to assign. The confidence targets of the assigned anchors are set to positive (one). Therefore, sample assignment is responsible for learning the confidence scores. We introduce LaneIoU to sample assignment in order to bring the detector’s confidence scores close to the segmentation-based IoU. LaneIoU dynamically determines the number of anchors to assign and prioritizes the anchors to assign as a cost function. Moreover, the IoU loss to regress the horizontal coordinates is also replaced by our LaneIoU to appropriately penalize the predicted lanes at different tilt angles. The LaneIoU integration to CLRNet [28] makes the detector’s training more straightforward, thus we coin our method *CLRerNet*.

We showcase the effectiveness of LaneIoU through extensive experiments on CULane and CurveLanes and report the state-of-the-art results on both datasets. Importantly, for a reliable and fair benchmark, we employ the average score

of five models for each experiment condition, while prior work shows a score of a single model. Moreover, the F1 metric employed in the lane detection evaluation is utterly sensitive to the detector’s lane confidence threshold, thus we determine the threshold utilizing the 5-fold cross validation on the train split.

Our contributions in this paper are threefold:

- *Clearer focus*: Through preliminary oracle experiments, we show that correct lane positions are already among the predictions of an existing detector, and the confidence scores that represent intersection-over-union (IoU) with ground truths are the most effective to improve performance.
- *Clearer training method*: As a lane similarity function, we leverage LaneIoU which well correlates with the evaluation metric and integrate it into training as a sample assignment cost and regression target.
- *Clearer benchmarking*: Multi-model evaluation and cross-validation-based score thresholding are employed for fair benchmark. The effectiveness and generality of LaneIoU is verified and CLRerNet achieves state-of-the-art in the CULane and CurveLanes benchmarks.

2. Related Work

2.1. Object detection

Training sample assignment. Sample assignment is the major research focus in object detection. The proposals from the detection head are assigned to the ground truth samples. [8, 12, 13, 22] assign the GTs by calculating IoU between the anchors on the feature map grid and GT boxes statically. [4] introduces the optimal transfer assignment (OTA) for object detector’s training sample assignment, that dynamically assigns the prediction boxes to the GTs. [5] simplifies OTA and realizes iteration-free assignment.

IoU functions. Several variants of IoU functions [23, 29, 30] are proposed for accurate bounding box regression and fast convergence. For example, the generalized IoU (GIoU) [23] introduces the smallest convex hull of the boxes and makes IoU differentiable even when the bounding boxes do not overlap. Our LaneIoU is based on GIoU but newly enables the IoU calculation between curves in the anchor-based and row-based methods.

2.2. Lane detection

Lane detection paradigms can be grouped as segmentation-based, keypoint-based, parametric, row-based and anchor-based methods.

Segmentation-based methods. This line of work is the bottom-up pixel-based estimation of lane existence proba-

bility. SCNN [18] and RESA [27] employ a semantic segmentation paradigm to classify the lane instances as separate classes on each pixel. The correspondence between lane and class is determined by annotation thus not flexible (e.g. some lane position may belong to two classes). The benchmark datasets [7, 18] employ pixel-level IoU to compare predicted lanes with GTs, and are friendly to the segmentation-based methods. However the methods do not treat lanes as holistic instances and require post-processing which is computationally costly. [19, 28] exploit the segmentation task as the auxiliary loss only during training time to improve the backbone network. We follow these methods and adopt the auxiliary branch and loss.

Keypoint-based methods. Similar to human pose estimation, the lane points are detected as keypoints and grouped into lane instances afterwards. PINet [11] employs test-time detachable stacked hourglass networks to learn keypoint probabilities and cluster the keypoints into the lane instances. FOLOLane [21] also detects lanes as keypoints inspired by the bottom-up human pose detection method [1]. GANet [10] regresses the offsets of the detected keypoints from the starting point of the corresponding lane instances. This line of methods also requires post-process to group the lane points into lane instances.

Parametric methods. In this line of work, a lane instance is represented as a set of curve parameters. PolyLaneNet [25] employs a curve representation using polynomial coefficients. LSTR [16] employs end-to-end transformer-based lane parameter set detection. BSNet [2] chooses the quasi-uniform b-spline curves and shows the highest F1 score among this category. These methods achieve relatively fast inference, however an error of one parameter holistically affects the lane shape.

Row-based methods. UFLD [19] captures the global features by flattening the feature map and learns row-wise lane position classification. UFLDv2 [20] extends [19] to a row- and column-wise lane representation to deal with the near-horizontal lanes. CondLaneNet [15] learns a probability heatmap of lane start points from where the dynamic convolution kernels are extracted. The dynamic convolution is applied to the feature map, whereby the row-wise lane point classification and x-coordinate regression are carried out. LaneFormer [6] employs a transformer with row and column attention to detect lane instances in an end-to-end manner. Additionally vehicle detection results are fed to the decoder to make the pipeline object-aware.

Anchor-based methods. The lane instance is represented as a set of x-coordinates at the fixed rows. LaneATT [24] employs lane anchors to learn the confidence score and local x-coordinate displacement for each anchor. An anchor is defined as a fixed angle and a start point. The training target is assigned statically according to the horizontal distance between each anchor to GTs. CLRNet [28] adopts learnable

anchor parameters (start point x_a, y_a and θ_a) and length l . For sample assignment, the simplified optimal transport assignment [5] is employed to dynamically allocate the closest predictions to each ground truth. Both methods pool the feature map by the anchors and feed the extracted features to the head network. The head network outputs the classification and regression tensors for each anchor. This paradigm corresponds to the 2-stage object detection methods such as [8, 13]. The anchor-based methods are the de-facto standard in terms of detection performance among the five method types.

3. Methods

3.1. Network design and losses

The anchor-based methods [24, 28] leverage the most accurate but simple detection pipeline among the five types described in Section 2. From the anchor-based methods we employ the best-performing CLRNet [28] as a baseline. The network schematic is shown in Fig. 2. The backbone network (e.g. ResNet [9] and DLA [26]) and the up-sampling network extract multi-level feature maps whose spatial dimensions are (1/8, 1/16, 1/32) of the input image. The initial anchors are formed from the N_a learnable anchor parameters (x_a, y_a, θ_a) where (x_a, y_a) is the starting point and θ_a the tilt of the anchor. The feature map is sampled along each of them and fed to the convolution and fully-connected (FC) layers. The classification logits c , anchor refinement $\delta x_a, \delta y_a, \delta \theta_a$, length l and local x-coordinate refinement δx tensors are output from the FC layers. The anchors refined by $\delta x_a, \delta y_a$ and $\delta \theta_a$ re-sample the higher-resolution feature map, and the procedure is repeated for three times. The pooled features are interacted with the feature map via cross-attention and are concatenated across different refinement stages. The lane prediction is expressed as classification (confidence) logits and a set of x-coordinates at N_{row} rows calculated from the final x_a, y_a, θ_a, l and δx . More details about the refinement mechanism can be found in [28]. During training, the predictions close to a GT are assigned via dynamic assigner [5]. The assigned predictions are regressed toward the corresponding GT and learned to be classified as positives.

$$L = \lambda_0 L_{reg} + \lambda_1 L_{cls} + \lambda_2 L_{seg} + \lambda_3 L_{LaneIoU} \quad (1)$$

where L_{reg} is smooth-L1 loss to regress the anchor parameters (x_a, y_a, θ_a) and l , L_{cls} a focal loss [14] for positive-or-negative anchor classification, L_{seg} an auxiliary cross-entropy loss for per-pixel segmentation mask, and $L_{LaneIoU}$ the newly introduced LaneIoU loss.

3.2. Oracle experiments

We conduct the preliminary oracle experiments to determine the direction of our approach. The prediction com-

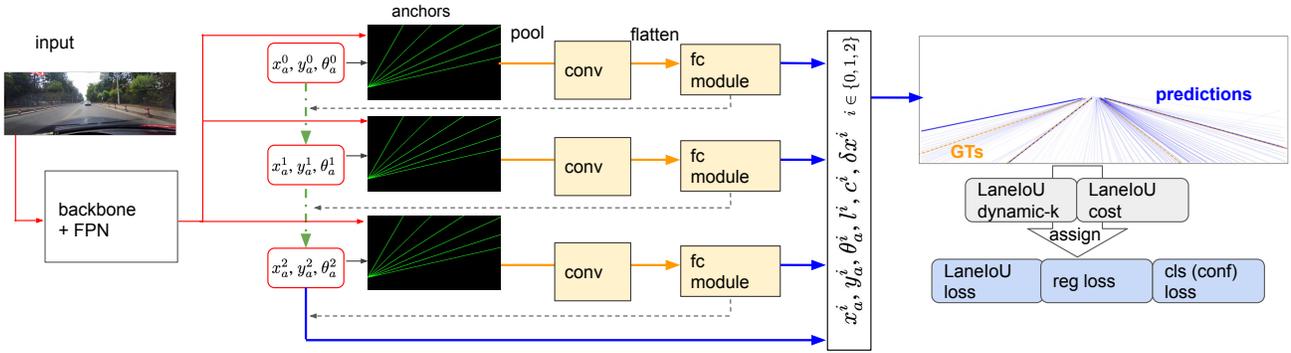


Figure 2. Network Schematic of CLRRerNet.

ponents from the trained baseline model are replaced with GTs partially to analyze how much room for improvement each lane representation component has. Table 1 shows the oracle experiment results. The baseline model (first row) is CLRRerNet-DLA34 trained without the redundant frames. The confidence threshold is set to 0.39 which is obtained via cross-validation (see Subsection 4.1 and 4.2).

Next, we calculate the metric IoU between predictions and GTs as the oracle confidence scores. For each prediction, the maximum IoU among the GTs is employed as the oracle score. In this case, the predicted lane coordinates are not changed. The $F1_{50}$ jumps to 98.47 - the near-perfect score (second row). The result suggests that **the correct lanes are already among the predictions, however the confidence scores need to be predicted accurately representing the metric IoU.**

The other components are the anchor parameters - x_a, y_a, θ_a and length l that determine the lane coordinates. We alter the anchor parameters and lane length by those of GTs (third and fourth rows respectively). Although the row-wise refinement δx is not changed, the oracle anchor parameters improve $F1_{50}$ by 9 points. On the other hand, the oracle length does not affect the performance significantly. The results lead to the second suggestion that the anchor parameters (x_a, y_a, θ_a) are important in terms of lane localization.

We focus on the first finding and aim to learn high-quality confidence scores by improving the lane similarity function.

3.3. LaneIoU

The existing methods [15] and [28] exploit horizontal distance and horizontal IoU as similarity functions respectively. However, these definitions do not match the metric IoU calculated with segmentation masks. For instance, when the lanes are tilted, the horizontal distance corre-

confidence	(x_a, y_a, θ_a)	l	$F1_{50}$
			80.86
✓			98.47
	✓		89.91
		✓	81.09

Table 1. Oracle experiment results.

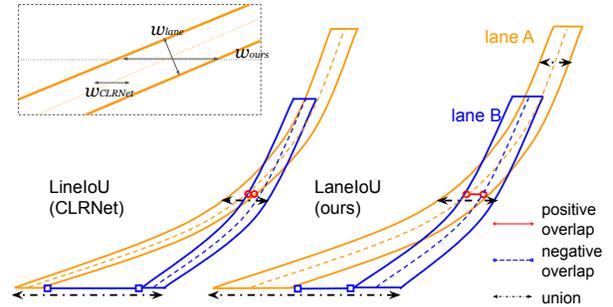


Figure 3. Example of LineIoU [28] and LaneIoU calculations between lane A and lane B. w_{lane}, w_{CLRNet} and w_{ours} within the dashed rectangle stand for the lane width, the constant width of [28] and our angle-aware width.

sponding to the certain metric-IoU is larger than that of vertical lanes. To bridge the gap, we introduce a differentiable local-angle-aware IoU definition, namely LaneIoU. Fig. 3 shows an example of IoU calculation between two tilted curves. We compare LineIoU [28] and LaneIoU on the same lane instance pair. LineIoU applies a constant virtual width regardless of the lane angle and the virtual lane gets 'thin' at the tilted part. In our LaneIoU, overlap and union are calculated considering the tilt of the local lane parts at each row. We define Ω_{pq} as the set of y slices where

both lanes p and q exist, and Ω_p or Ω_q where only one lane exists. LaneIoU is calculated as:

$$LaneIoU = \frac{\sum_{i=0}^H I_i}{\sum_{i=0}^H U_i}, \quad (2)$$

where I_i and U_i are defined as:

$$I_i = \min(x_i^p + w_i^p, x_i^q + w_i^q) - \max(x_i^p - w_i^p, x_i^q - w_i^q), \quad (3)$$

$$U_i = \max(x_i^p + w_i^p, x_i^q + w_i^q) - \min(x_i^p - w_i^p, x_i^q - w_i^q), \quad (4)$$

when $i \in \Omega_{pq}$. The intersection of the lanes is positive when the lanes are overlapped and negative otherwise.

If $i \notin \Omega_{pq}$, I_i and U_i are calculated as follows:

$$I_i = 0, U_i = 2w_i^k \text{ if } k \in \{p, q\}, i \in \Omega_k, \quad (5)$$

$$I_i = 0, U_i = 0 \text{ if } i \notin (\Omega_{pq} \cup \Omega_p \cup \Omega_q). \quad (6)$$

The virtual lane widths w_i^p and w_i^q are calculated taking the local angles into consideration:

$$w_i^k = \frac{w_{lane}}{2} \frac{\sqrt{(\Delta x_i^k)^2 + (\Delta y_i^k)^2}}{\Delta y_i^k}, \quad (7)$$

where $k \in \{p, q\}$ and Δx_i and Δy_i stand for the local changes of the lane point coordinates. Equation 7 compensates the tilt variation of lanes and represents a general row-wise lane IoU calculation. When the lanes are vertical, w_i equals to $w_{lane}/2$ and gets larger as the lanes tilt. w_{lane} is the parameter which controls the strictness of the IoU calculation. The CULane metric employs 30 pixels for the resolution of (590, 1640).

In Fig. 4, LineIoU [28] and our LaneIoU are compared by calculating correlation with the CULane metric. We replace each prediction's confidence score with the LineIoU or LaneIoU value and also calculate the metric IoU. The GT with the largest IoU is chosen for each prediction. Clearly our LaneIoU shows better correlation with the metric IoU mainly as the result of eliminating the influence of lane angles.

3.4. Sample assignment

The confidence scores are learned to be high if the anchor is assigned as positive during training. We adopt LaneIoU for sample assignment to bring the detector's confidence scores close to the segmentation-based IoU. [28] employs the SimOTA assigner [5] to dynamically assign k_i anchors for each GT lane t_i . The number of anchors k_i is determined by calculating the sum of all the anchors' positive IoUs. We employ LaneIoU as:

$$k_i = \sum_{j=1}^m LaneIoU(p_j, t_i), \quad (8)$$

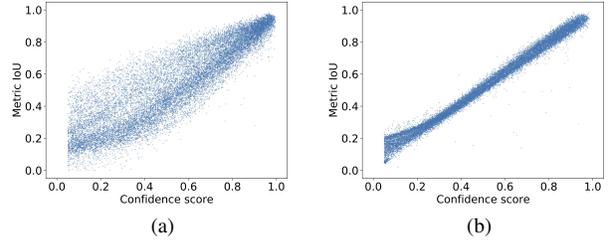


Figure 4. Correlation between CULane metric IoU and (a) LineIoU [28] and (b) LaneIoU (ours).

where m is the number of anchors and $i = 0, 1, \dots, n$ is the index of n GT lanes. p_j is a predicted lane and $j = 0, 1, \dots, m$ is the index of m predictions. k_i is clipped to be from 1 to k_{max} . The cost matrix determines the priority of the assignment for each GT. In object detection, [5] adopts the sum of bounding box IoU and classification costs. In lane detection, CLRNet [28] utilizes the classification cost and the lane similarity cost which consists of horizontal distance, angle difference and start-point distance. However, the cost that directly represents the evaluation metric is more straightforward for prioritizing predictions. We define the cost matrix as:

$$cost_{ji} = -LaneIoU_{norm}(p_j, t_i) + \lambda f_{class}(p_j, t_i), \quad (9)$$

where λ is the parameter to balance the two costs and f_{class} is the cost function for classification, such as a focal loss [14]. LaneIoU is normalized from its minimum to maximum. The formulation (eq. 8 and 9) realizes dynamic sample assignment of the proper number of anchors which are prioritized according to the evaluation metric. In summary, compared with CLRNet, our CLRRerNet introduces LaneIoU as the dynamic-k assignment function, assignment cost function and loss function to learn the high-quality confidence scores that better correlate with the metric IoU.

4. Experiments

4.1. Datasets

The CULane dataset¹ [18] is the de-facto standard lane detection benchmark dataset which contains 88,880 train frames, 9,675 validation frames, and 34,680 test frames with lane point annotations. The test split has frame-based scene annotations such as Normal, Crowded and Curve (see Table 2). The CurveLanes² [7] dataset contains the challenging curve scenes and consists of 100k train, 20k val and 30k test frames. We follow [15] and use the val split for evaluation.

¹<https://xingangpan.github.io/projects/CULane.html>

²<https://github.com/SoulmateB/CurveLanes>

Removing the redundant train data. The CULane dataset includes a non-negligible amount of redundant frames where the ego-vehicle is stationary and the lane annotations do not change. We have found that overfitting to the redundant frames can be avoided by simply removing the frames whose average pixel value difference from the previous frame is below a threshold. The optimal threshold (=15) is chosen empirically via validation as described in the supplementary material. The remaining 55,698 (62.7%) frames are utilized for training. The F1 score of CLRNNet-DLA34 is improved from 80.30 ± 0.05 to 80.86 ± 0.06 ($N = 5$ each) with the same 15-epoch training.

4.2. Training and evaluation

The models are implemented on PyTorch and MMDetection [3], and are trained for 15 epochs with AdamW [17] optimizer. The initial learning rate is 0.0006 and cosine decay is applied. For CULane dataset, we crop the input image below $y = 270$ and resize it to (800, 320) pixels. Horizontal flip, random brightness and contrast, random HSV modulation, motion and median blur and random affine modulations are adopted as data augmentation, following [28]. At the test time only the crop and resize are adopted and no test-time augmentations are applied. In CLRNerNet, LaneIoU is introduced as a loss function, dynamic-k calculation and assignment loss function. w_{lane} is set to 15/800 for loss and dynamic-k, and 60/800 for cost to balance with the classification cost. The loss weights in eq. 1 are the same as [28] except for λ_3 which is set to 4. We additionally benchmark a CLRNerNet-DLA34 model trained for 60 epochs applying exponential moving average (EMA). The learning rate decay is not applied and the momentum of EMA is set to 0.0001.

To validate the generality of our method, we add the LaneIoU-based sample assignment to LaneATT [24]. Originally, LaneATT assigns non-learnable static anchors to GTs by horizontal distance thresholding. We prioritize the anchors by calculating LaneIoU between predicted lanes and GTs to assign the positive-confidence targets. More details are described in the supplementary material.

For CurveLanes [7], we follow the training setting of [15] where the input resolution is (800, 320). To exploit the auxiliary segmentation loss, we draw the segmentation mask along all the lane labels with width of 30 pixels. Different from [18], we set all the lane masks as class one (foreground). Since the test annotations are not available, we evaluate our method on the validation split. We employ the evaluation resolution of (224, 224) and line width of 5 following [15]. w_{lane} is set to 5/224 for loss and dynamic-k calculation and 20/224 for cost. λ for assignment cost calculation (eq. 9) is set to 2.5.

Evaluation metric. We employ F1 score [18] as an evaluation metric. An IoU matrix between predicted lanes and

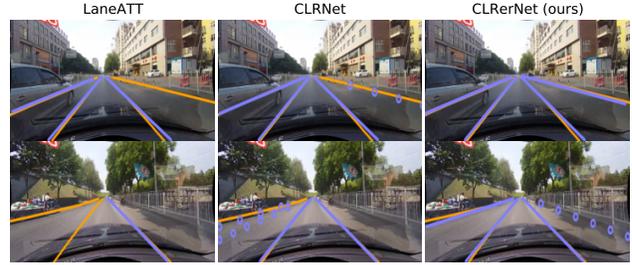


Figure 5. Qualitative results comparing LaneATT, CLRNNet and our CLRNerNet†*. Predictions and GTs are shown in blue and orange respectively. Predictions with insufficient confidence score are shown as blue circles.

ground-truths is calculated by comparing the segmentation masks drawn with a width of 30 pixels (Fig. 1 bottom). Based on the IoU matrix, one-to-one matching is calculated using linear sum assignment and the prediction-GT pairs with IoU over t_{IoU} are considered as true positives (TP). Unmatched predictions and GTs are counted as false positives (FP) and false negatives (FN) respectively. We employ two t_{IoU} values for IoU calculation: 0.5 and 0.75. The F1 score is calculated as:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (10)$$

Cross validation and test. The F1 metric is sensitive to the threshold of detection confidence score. We perform 5-fold cross validation on the train split, by randomly dividing the train videos into five groups. The F1 score of each threshold is averaged across the 5-fold results, and the optimal threshold is determined by taking the argmax of it.

Moreover, we find that the F1 score deviation of the models trained with different random seeds is not negligible. For instance, the F1 score of the CLRNNet-DLA34 ranges from the minimum of 80.20 to the maximum of 80.34 ($N = 5$). For a more reliable and fairer benchmark on the test split, we train five models with different random seeds for each condition and calculate the mean and standard deviation of the metrics, on which the confidence thresholds obtained by five-fold cross validation is applied. To the best of our knowledge, we are the first to conduct the above benchmark protocol in lane detection. As for the CurveLanes dataset where the test annotations are not available, we report the maximum F1 score on the validation split with respect to the confidence thresholds.

4.3. CULane benchmark results

The benchmark results on the CULane test set are shown in Table 2. The rows below the double horizontal line are our experiment results. For each condition (row), we show

Method	Backbone	F1 ₅₀	F1 ₇₅	Normal	Crowd	Dazzle	Shadow	Noline	Arrow	Curve	Cross	Night	GFLOPs	FPS
SCNN	VGG16	71.60	39.84	90.60	69.70	58.50	66.90	43.40	84.10	64.40	1990	66.10	218.6	50
LaneATT	Res18	75.13	51.29	91.17	72.71	65.82	68.03	49.13	87.82	63.75	1020	68.58	9.3	211
LaneATT	Res34	76.68	54.34	92.14	75.03	66.47	78.15	49.39	88.38	67.72	1330	70.72	18.0	170
CondLane [15]	Res18	78.14	57.42	92.87	75.79	70.72	80.01	52.39	89.37	72.40	1364	73.23	10.2	348
CondLane [15]	Res34	78.74	59.39	93.38	77.14	71.17	79.93	51.85	89.89	73.88	1387	73.92	19.6	237
CondLane [15]	Res101	79.48	61.23	93.47	77.44	70.93	80.91	54.13	90.16	75.21	1201	74.80	44.8	97
CLRNet [28]	Res34	79.73	62.11	93.49	78.06	74.57	79.92	54.01	90.59	72.77	1216	75.02	21.5	204
CLRNet [28]	Res101	80.13	62.96	93.85	78.78	72.49	82.33	54.50	89.79	75.57	1262	75.51	42.9	94
CLRNet [28]	DLA34	80.47	62.78	93.73	79.59	75.30	82.51	54.58	90.62	74.13	1155	75.37	18.4	185
LaneATT†	Res34	77.51±0.10	56.78	92.48	75.47	68.09	73.21	50.96	88.72	68.18	1054	72.58	18.0	170
CLRNet†	Res34	80.54±0.12	63.65	93.85	79.22	73.32	82.50	55.26	90.84	74.06	1106	75.92	21.5	204
CLRNet†	Res101	80.67±0.06	64.35	93.95	79.60	72.91	81.58	55.76	90.42	74.06	1166	76.01	42.9	94
CLRNet†	DLA34	80.86±0.06	64.05	94.03	79.78	75.23	81.94	56.02	90.67	74.57	1184	76.40	18.4	185
LaneATT+†	Res34	78.19±0.06	56.96	92.60	76.42	69.12	77.59	52.01	88.75	64.49	974	72.78	18.0	153
CLRerNet†	Res34	80.76±0.13	63.77	93.93	79.51	73.88	83.16	55.55	90.87	74.45	1088	76.02	21.5	204
CLRerNet†	Res101	80.91±0.10	64.30	93.91	80.03	72.98	82.92	55.73	90.53	73.83	1113	76.13	42.9	94
CLRerNet†	DLA34	81.12±0.04	64.07	94.02	80.20	74.41	83.71	56.27	90.39	74.67	1161	76.53	18.4	185
CLRerNet†*	DLA34	81.43±0.14	65.06	94.36	80.62	75.23	84.35	57.31	91.17	79.11	1540	76.92	18.4	185

Table 2. Evaluation results on the CULane test set. Our experiments are below the double horizontal line.

the averaged metric values of five models trained with different seeds. The confidence threshold obtained from 5-fold cross-validation is employed. The F1₅₀ scores are shown in the test scene columns except for the *cross* metric where the number of false positives is shown. All the FPS results on Table 2 are measured with a GeForce RTX 3090 GPU. CLRNet† and LaneATT† are the baseline model trained with our implementation. Our method CLRerNet† employs LaneIoU for dynamic-k calculation, assignment cost and loss functions. CLRerNet†* is the boosted version of CLRerNet† which is trained for 60 epochs with EMA.

With introducing LaneIoU, CLRerNet† with DLA34 outperforms CLRNet† by 0.26% in F1₅₀. Moreover, the boosted model CLRerNet†* reaches F1₅₀ = 81.43% in average, enjoying the state-of-the-art performance surpassing the previous methods (F1₅₀ = 80.47%, single experiment of CLRNet+DLA34) by a large margin. The performance improvement by LaneIoU is also observed on the models with other backbones - 80.54% to 80.76% (+0.22%) with ResNet34 and 80.67% to 80.91% (+0.24%) with ResNet101. LaneATT+† is improved by the LaneIoU-based assignment by 0.68%, validating the generality of our method. CLRerNet does not increase test-time computational complexity and shows the same GFLOPs and FPS as CLRNet.

Qualitative results on the CULane test set are shown in Fig. 5. Our CLRerNet†* is capable of detecting the lanes in the challenging scenes. The right-most tilted lane of the first image (top) and the left-most lane of the second image (bottom) are detected only by CLRerNet with high confidence scores. The examples qualitatively suggest that CLRerNet is able to give more correct scores to predictions, which is analyzed in Subsection 4.5.

Method	F1	Precision	Recall	GFLOPs
CondLane-S [15]	85.09	87.75	82.58	10.3
CondLane-M [15]	85.92	88.29	83.68	19.7
CondLane-L [15]	86.10	88.98	83.41	44.9
CLRNet-DLA34 [28]	86.10±0.08	91.40	81.39	18.4
CLRerNet-DLA34	86.47±0.07	91.66	81.83	18.4

Table 3. Comparison between methods on the CurveLanes *val* set. Our experiments are below the double horizontal line.

4.4. CurveLanes validation results

The validation results on CurveLanes are shown in Table 3. The default CLRNet [28] with the DLA34 backbone shows the same F1 score as CondLane-L [15] with lower computation cost. Note that our results are the average of five training trials. The confidence threshold is set to the empirically optimal value 0.44. CLRerNet significantly outperforms the baseline by 0.37%, achieving the new state-of-the-art 86.47%.

4.5. Ablation study

We corroborate the effectiveness of our method by ablating LaneIoU from dynamic-k calculation, assignment cost and loss function. CLRerNet with DLA34 backbone is trained in each condition with the redundant train data omitted. We follow the benchmark protocol described in subsection 4.2, thus ten models (5 seeds + 5 folds) are trained and validated for each condition. The results in Table 4 show that the performance degrades by replacing LaneIoU with [28] for dynamic-k determination, cost function and loss function respectively. Determining the number of assignments each GT lane by LaneIoU mitigates the inhomogeneity caused by lane tilt variation. The LaneIoU-based assignment cost (eq. 9) prioritizes the predicted lanes which

dynamic-k	cost	loss	F1 ₅₀	F1 ₇₅
[28]	[28]	[28]	80.86±0.06	64.05±0.17
LaneIoU	[28]	[28]	80.98±0.07	64.17±0.17
LaneIoU	LaneIoU	[28]	81.07±0.03	64.22±0.26
LaneIoU	LaneIoU	LaneIoU	81.12±0.05	64.28±0.15

Table 4. Ablation study by replacing LaneIoU (ours) with [28].

have higher metric IoU with the GTs, leading to more accurate confidence learning as motivated in subsection 3.2. Replacing LineIoU loss with LaneIoU loss also mitigates the tilt dependency of the regression penalty.

In summary, F1 score improvement solely by LaneIoU is +0.26%. On the other hand, the scores with Res34, 101 and DLA34 backbones are 80.54, 80.67 and 80.86 % (see Table 2) where each improvement is less than 0.20%. Res101 has 2× GFLOPs than Res34 but shows only 0.13% F1 difference. Therefore, LaneIoU’s improvement is twice as significant as the double-GFLOPs difference.

4.6. Analysis

Fig. 6 shows the comparison between CLRerNet and CLRNet in terms of anchor assignment numbers per GT in different angle ranges. The assignment numbers are accumulated during the training and averaged. The angles are calculated using GT lanes in (800, 320) resolution and 90° corresponds to the vertical lane. CLRNet wrongly assigns more anchors to vertical GTs and few for near-horizontal ones (by underestimating IoU), making training suboptimal. LaneIoU mitigates the issue, homogeneously assigning anchors and distributing losses to diverse-angle GTs, especially in the angle ranges of 20° to 60° and 120° to 160° where the GTs typically exist.

The assigned anchor’s confidence target is set to positive prioritized by LaneIoU. Therefore, the confidence is trained more homogeneously across different lane angles. As can be seen in Fig. 7, the l_1 error between the predicted confidence scores and the metric IoU values is improved in CLRerNet in the non-vertical angle ranges, corroborating the effectiveness of LaneIoU.

4.7. Limitations

Although CLRerNet shows significant improvement in performance, there still is a gap between the best CLRerNet model’s performance (81.43%) and the oracle-confidence case (98.47%). The dataset-oriented issues including label fluctuation and data imbalance are considered to be the part of the gap. As can be found in Table 2, the *Noline* test category is the most challenging as there are no visual markings on the road. Such cases are prone to label fluctuation and inconsistency. Likewise, data imbalance such as stationary scenes greatly affects the model training. As is mentioned

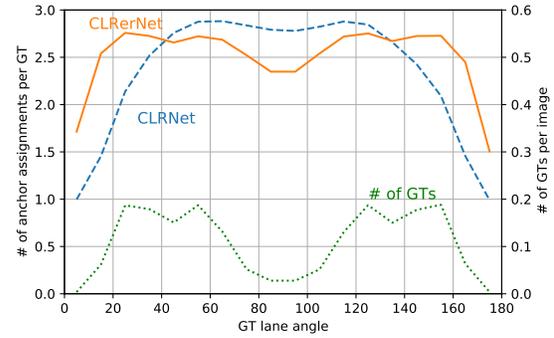


Figure 6. The average number of assignments in different angle ranges.

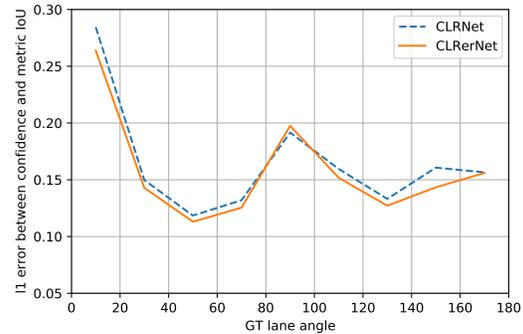


Figure 7. The average l_1 error between confidence score and metric IoU in different angle ranges.

in Subsection 4.1, we find that mitigating the data imbalance significantly improves the performance.

5. Conclusion

We disentangle the lane prediction components by the oracle experiment and demonstrate the importance of high-quality confidence scores for more accurate lane detection. To make confidence scores represent the metric IoU, the novel LaneIoU is proposed and integrated into the anchor-based lane detection baselines. A novel detector coined CLRerNet is developed by introducing LaneIoU as the sample assignment and loss functions. The statistical and fair benchmark protocol is employed utilizing five-seed models and five-fold cross validation. CLRerNet achieves the state-of-the-art performance on the challenging CULane and CurveLanes datasets significantly surpassing the baseline. We believe our oracle experiments, LaneIoU-based training and benchmark protocol bring a clearer view of lane detection to the community.

References

- [1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3
- [2] Haoxin Chen, Mengmeng Wang, and Yong Liu. Bsnet: Lane detection via draw b-spline curves nearby. *arXiv preprint arXiv:2301.06910*, abs/2301.06910, 2023. 3
- [3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6
- [4] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 303–312, June 2021. 2
- [5] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 2, 3, 5
- [6] Jianhua Han, Xiajun Deng, Xinyue Cai, Zhen Yang, Hang Xu, Chunjing Xu, and Xiaodan Liang. Laneformer: Object-aware row-column transformers for lane detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):799–807, Jun. 2022. 3
- [7] Xinyue Cai, Wei Zhang, Xiaodan Liang, Zhenguo Li, Hang Xu, Shaoju Wang. Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In *ECCV*, 2020. 1, 3, 5, 6
- [8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 2, 3
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3
- [10] Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, Tianzhu Zhang, Jinsheng Wang, Yinchao Ma. A keypoint-based global association network for lane detection. In *CVPR*, 2022. 3
- [11] Yeongmin Ko, Younkwon Lee, Shoaib Azam, Farzeen Munir, Moongu Jeon, and Witold Pedrycz. Key points estimation and point instance segmentation approach for lane detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8949–8958, 2022. 3
- [12] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 2
- [13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 2, 3
- [14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3, 5
- [15] 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 (ICCV)*, pages 3773–3782, October 2021. 1, 3, 4, 5, 6, 7
- [16] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3693–3701, 2021. 1, 3
- [17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6
- [18] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*, February 2018. 1, 3, 5, 6
- [19] Zequn Qin, Huanyu Wang, and Xi Li. Ultra fast structure-aware deep lane detection. In *The European Conference on Computer Vision (ECCV)*, 2020. 1, 3
- [20] Zequn Qin, Pengyi Zhang, and Xi Li. Ultra fast deep lane detection with hybrid anchor driven ordinal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–14, 2022. 1, 3
- [21] Zhan Qu, Huan Jin, Yang Zhou, Zhen Yang, and Wei Zhang. 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 (CVPR)*, pages 14122–14130, June 2021. 1, 3
- [22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, volume 28. Curran Associates, Inc., 2015. 2
- [23] Hamid Rezaatoughi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2
- [24] Lucas Tabelini, Rodrigo Berriel, Thiago M. Paixão, Claudine Badue, Alberto Ferreira De Souza, and Thiago Oliveira-Santos. Keep your Eyes on the Lane: Real-time Attention-guided Lane Detection. In *CVPR*, 2021. 1, 3, 6
- [25] Lucas Tabelini Torres, Rodrigo Ferreira Berriel, Thiago M. Paixão, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. PolyLaneNet: Lane estimation via deep polynomial regression. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 6150–6156. IEEE, 2020. 1, 3
- [26] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3
- [27] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(4):3547–3554, May 2021. 1, 3

- [28] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 898–907, June 2022. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [29] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2020. [2](#)
- [30] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. 2021. [2](#)