RCLane: Relay Chain Prediction for Lane Detection

Shenghua Xu[1⋆], Xinyue Cai[2⋆], Bin Zhao[1], Li Zhang[1⋆⋆], Hang Xu[2], Yanwei Fu[1], and Xiangyang Xue[1]

[1] Fudan University
[2] Huawei Noah's Ark Lab

1 Experimental setting for step length d

Step length d is the distance between the two neighbors when encoding the lane, which is fixed as 10 for all the experiments in our method. LaneATT [7] sets the line width as 30 for calculating IoU. We set it as 15 according to the resolution scale initially. And d should be at least half of the line width to ensure all foreground points find neighbors sited at the center line. Thus we set d as 7, 8, 9, 10, 11 and 12 and conduct a series of experiments on CurveLanes [12]. The quantitative results are shown in Tab. 1. We find that F1-score decreases as d increases since small step length can describe the local variable shape of lanes more precisely while increasing decoding time. 10 is chosen as the final setting considering the performance-speed trade-off.

Table 1. Ablation study on step length on CurveLanes.

Step length	Precision(%)	Recall(%)	F1(%)
7	94.29	88.87	91.50
8	94.26	88.84	91.47
9	94.28	88.79	91.45
10	93.96	89.03	91.43
11	93.93	88.97	91.38
12	94.16	88.93	91.31

2 Attention mechanisms

We consider the recent attention mechanism is suitable for information processing and do ablation experiments on *CurveLanes*. Results are shown in Tab. 2. Adding self-attention [9] to U-Net [6], which operated on the deepest feature

⋆ Equal contribution.
⋆⋆ Li Zhang (lizhangfd@fudan.edu.cn) is the corresponding author with School of Data Science, Fudan University.

map, makes F1-score increase from 89.41% and reaches 89.49%. In the third row, we replace it with axial attention [10] and further improves the performance considering the row-column style attention adapts to the long and thin characteristics of lanes. Finally, we utilize the efficient transformer-based network SegFormer as backbone and achieve the best result. From the above results, we can find attention mechanism can help our model focus on local location and global shape information simultaneously.

Table 2. Comparison among different settings of attention on the CurveLanes. The attention module is just added to the feature map of the smallest resolution.

Backbone	Precision(%)	Recall(%)	F1(%)
Unet [6]	93.11	86.00	89.41
Unet [6] + self-attention [9]	92.95	86.27	$89.49^{\mathbf{+0.08}}$
Unet [6] + axial-attention [10]	92.79	88.41	$90.55^{\mathbf{+1.14}}$
SegFormer-B2 [11]	93.96	89.03	$91.43^{\mathbf{+2.02}}$

3 Generalization

This section aims to verify the generalization capacity of our RCLane following FOLOLane [5]. We utilize the model trained on the CULane [3] training set to evaluate on the TuSimple [8] test set. The results are shown in Tab. 3. Our RCLane surpasses FOLOLane [5] by 2.88% , with smaller FP and FN indicating that our method is more robust than previous lane detection methods.

Table 3. Evaluation of generalization ability of different methods from CULane training set to TuSimple testing set.

Method	Accuracy(%)	FP(%)	FN(%)
PINet [2]	36.31	48.86	89.88
UFNet [4]	65.53	56.80	65.46
FOLOLane [5]	84.36	39.64	38.41
RCLane (Ours)	**87.24**	**22.06**	**21.56**

4 Visualization results on CULane and CurveLanes

The visualizations of detection results on CULane [3] and CurveLanes [12] are shown in Fig. 1. Our RCLane can perform well not only in easy scenes, but also in some complex scenes, for example, Fork-shape lanes and dense curve lanes.

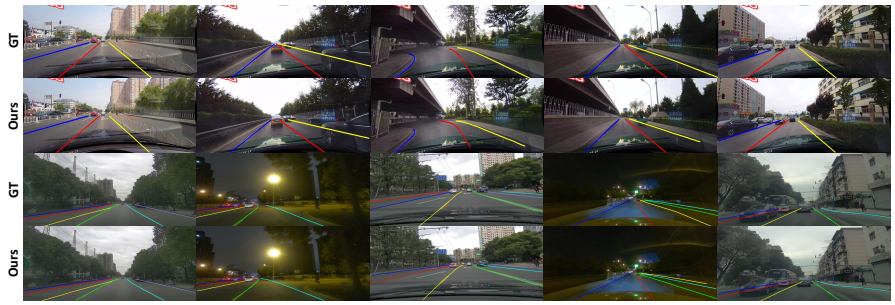

Fig. 1. Visualization results of our RCLane. The first two rows are of CULane and the last two rows are of CurveLanes. Our model can detect curve lanes, dense lanes and Y-shape lanes in different scenarios.

5 Visualization results on Tusimple and LLAMAS

The qualitative results on TuSimple [8] and LLAMAS [1] are shown in Fig. 2. TuSimple and LLAMAS are two benchmarks taken from the highway driving scenarios and are easier compared with CULane [3] and CurveLanes [12]. In some scenarios such as curve lanes or the far end of lanes, our RCLane even shows better performance than ground truths, as is shown in the last row and forth column in Fig. 2.

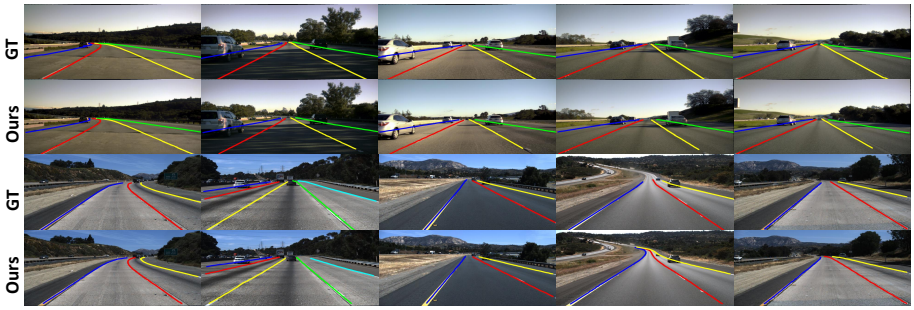

Fig. 2. Visualization on Tusimple and LLAMAS. The first two rows are the ground truth and our predictions on LLAMAS and the last two rows are the ground truth and our predictions on Tusimple.

6 Comparison of results on complex scenes

We choose some complex scenes of CurveLanes [12] and show the visualizations of different methods on Fig. 3. It shows the capacity of our RCLane on complex topologies with the global shape and local location information learning, while other methods fail to detect all lane instances when meeting Fork-shape lanes and Y-shape lanes. Moreover, we additionally visualize more results of our RCLane on other hard cases in Fig. 4 and achieve good performance.

Fig. 3. Comparison of visualization results between our RCLane and other lane detection methods. Our method performs better in different scenarios such as occluded, curved and Y-shape lanes.

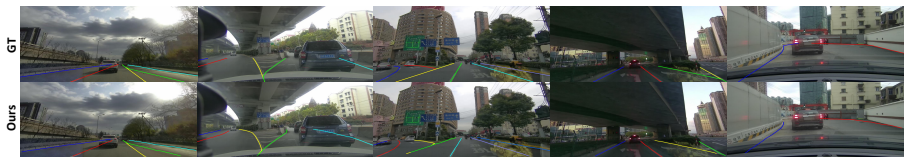

Fig. 4. Visualization of detection results for complex topologies, such as fork-shape, Y-shape and nearly horizontal lanes, shows the capacity of RCLane.

7 Pseudo code

7.1 Pseudo code for lane encoder

Algorithm·1 details the process of lane encoder, which aims to generate the ground truth of the transfer map and distance map to supervise the training process.

Algorithm 1 Lane encoder

Input: d: the step length
$\overline{S}$: the foreground point set in segmentation map
L: the set of lanes in the image I
$g_1(\cdot)$: the function to compute the distance between a point and a lane
$g_2(\cdot)$: the function to compute the distance between two points
Output: $\overline{T}_f$: the forward transfer map
$\overline{T}_b$: the backward transfer map
$\overline{D}_f$: the forward distance map
$\overline{D}_b$: the backward distance map

for all $p_i = (x_i, y_i) \in \overline{S}$ **do**
  $l = \arg\min_{\gamma \in L} g_1(p_i, \gamma)$
  if $g_1(p_i, l) < d$ **then**
    find the forward and backward end points of l:
    $p^f_{end} = (x^f_{end}, y^f_{end}) = \arg\max_{(x,y)\in l}(y)$,
    $p^b_{end} = (x^b_{end}, y^b_{end}) = \arg\min_{(x,y)\in l}(y)$
    compute the forward and backward distance scalars:
    $\overline{D}_f(p_i) = \sqrt{(x_i - x^f_{end})^2 + (y_i - y^f_{end})^2}$,
    $\overline{D}_b(p_i) = \sqrt{(x_i - x^b_{end})^2 + (y_i - y^b_{end})^2}$
    find the forward and backward point for p_i on lane l:
    $p^f_i = (x^f, y^f) = \arg\max_{g_2((x,y),p_i)=d}(y)$,
    $p^b_i = (x^f, y^f) = \arg\min_{g_2((x,y),p_i)=d}(y)$
    compute the two transfer vectors:
    $\overline{T}_f(p_i) = (x^f_i - x_i, y^f_i - y_i)$,
    $\overline{T}_b(p_i) = (x^b_i - x_i, y^b_i - y_i)$
  end if
end for
return $\overline{T}_f$,$\overline{T}_b$,$\overline{D}_f$ and $\overline{D}_b$

7.2 Pseudo code for lane decoder

Algorithm 2 introduces the detail process for lane decoder, showing how to recover all possible lanes based on the predicted segmentation map, transfer map and distance map.

Algorithm 2 Lane decoder

Input: d : the step length
r: the minimal distance between two foreground points in key point map
$f(\cdot, r)$: the function to perform Point-NMS on segmentation map to get a sparse key point map
$h(\cdot)$: the function to perform IOU-NMS on the predict set of lanes
S: the predicted segmentation map
T: the predicted transfer map
D: the predicted distance map
Output: L: the set of lanes in the image I

key point map $K = f(S, r)$
initialize L' as an empty set
for all key point $p \in K$ **do**
  initialize two point sets: $G_1 = \{p\}$ and $G_2 = \{p\}$
  initialize two points: $p_1 = p$ and $p_2 = p$
  for $i = 1$ to $\frac{D_f(p)}{d}$ **do**
    get the forward transfer vector $T_f(p_1)$ from T_f
    update p_1: $p_1 = p_1 + T_f(p_1)$
    update G_1: $G_1 = G_1 \cup \{p_1\}$
  end for
  for $i = 1$ to $\frac{D_b(p)}{d}$ **do**
    get the backward transfer vector $T_b(p_2)$ from T_b
    update p_2: $p_2 = p_2 + T_b(p_2)$
    update G_2: $G_2 = G_2 \cup \{p_2\}$
  end for
  get the lane $l_p = \{G_1\} \cup \{G_2\}$
  update L': $L' = L' \cup \{l_p\}$
end for
the final results of predicted lanes $L = h(L')$
return the set of lanes L

References

1. Behrendt, K., Soussan, R.: Unsupervised labeled lane markers using maps. In: ICCV workshops (2019)
2. 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)
3. Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In: AAAI (2018)

4. Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection. In: ECCV (2020)
5. Qu, Z., Jin, H., Zhou, Y., Yang, Z., Zhang, W.: Focus on local: Detecting lane marker from bottom up via key point. In: CVPR (2021)
6. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
7. 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: CVPR (2021)
8. TuSimple: Tusimple benchmark. `https://github.com/TuSimple/tusimple-benchmark` (2019)
9. Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
10. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: ECCV (2020)
11. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint (2021)
12. Xu, H., Wang, S., Cai, X., Zhang, W., Liang, X., Li, Z.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: ECCV (2020)