CaT: Cyclic-Accumulation Transformer for Lane Detection

1st Dezhen Qi[1]
Xiamen University
Xiamen, China
lucky7dz@stu.xmu.edu.cn

2nd Jun Xie[3]
PCIE Lab, Lenovo Research
Beijing, China
xiejun@lenovo.com

3rd Guoyu Yang[1]
Xiamen University
Xiamen, China
ooygyoo@stu.xmu.edu.cn

4th Dongdong Liu[1]
Xiamen University
Xiamen, China
1151902112@qq.com

5th Ye Qiu[1]
Xiamen University
Xiamen, China
qiuye@stu.xmu.edu.cn

6th Yuer Lu[1]
Xiamen University
Xiamen, China
yuerlu@stu.xmu.edu.cn

7th Xiaoming Jiang[2]
Wenzhou Institute, University of Chinese Academy of Sciences
Wenzhou, China
jiangvic2021@163.com

8th Jianwei Shuai*[12]
Xiamen University
Xiamen, China
jianweishuai@xmu.edu.cn

2023 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-8867-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/IJCNN54540.2023.10191407

***Abstract*—Lane detection is a special task in autonomous driving. Its most prominent inherent feature is to learn the imagination of severely occluded objects. Traditional CNN-based networks learning the imagination tend to perform poorly. In this work, we propose a novel architecture, called Cycle_accumulation-Transformer (CaT), which is the first structure to handle the lane detection by fusing CNN and Transformer. In particular, Cycle_accumulation structure and Transformer structure complement each other, and they adopt the four-direction cyclic accumulation process of "up to down", "down to up", "left to right" and "right to left" in the convolutional mode and the self-attention mechanism of "QKV" to fuse global information respectively. Our method is based on pixel-level semantic segmentation with high detection accuracy while meeting real-time requirements. Moreover, our proposed method achieves state-of-the-art results on the Tusimple and also achieves competitive results on the CULane.**

***Index Terms*—lane detection, cycle_accumulation structure, Transformer, global information,**

I. Introduction

Lane line is a high-level visual language symbol defined in human society, which specifies the basic norms for vehicles to drive on the road. Lane detection plays an important role in both assisted driving and automatic driving, including: high-precision map generation, lane keeping during driving, automatic cruise and overtaking decision-making. The task of lane detection is to segment and detect the lane line contained in the 2D image captured by the vehicle camera, and it needs to meet certain accuracy and real-time requirements. The difficulties in lane detection can be roughly divided into the following two points: 1. Severe occlusion: Originating from crowded traffic; 2. The characteristics of the lane line itself: the lane line itself is a slender object, wear and tear, dotted or solid lines, etc.

*Correspoonding Author
[1]Department of Physics, and Fujian Provincial Key Laboratory for Soft Functional Materials Research, Xiamen University, Xiamen 361005, China
[2]Wenzhou Institute, University of Chinese Academy of Sciences, and Oujiang Laboratory (Zhejiang Lab for Regenerative Medicine, Vision and Brain Health), Wenzhou, Zhejiang 325001, China
[3]PCIE Lab, Lenovo Research, Beijing, 100089, China

Lane detection is a proprietary problem in a specialized field, its development benefits from the advancement of computer vision technology in a large environment. In the early days, traditional image algorithms [36] analyzed and processed images by finding the correlation between pixels and fusing geometric features. Although these algorithms were simple to implement, they were often inaccurate and could only be used in specific occasions.

In recent years, with the substantial improvement of hardware (GPU) computing power, deep learning methods have begun to enter the stage of history. In the field of computer vision (CV), the CNN network was first proposed in [11] in 1998, and scholars from all walks of life have successively put forward excellent paper ideas such as AlexNet [10], VGG [27], GoogLeNet [30], and ResNet [6]. The extremely creative work has broken through the bottleneck of previous traditional image algorithms, and has achieved a speed and accuracy comparable to that of human naked eye recognition in classification, segmentation, and detection tasks. CNN-based lane detection also achieves unprecedented real-time and accuracy requirements, and is applied to the complex system engineering of autonomous driving.

Transformer [33] is widely used in the field of Natural Language Processing (NLP), and some classic networks [2], [23], [24] contain Transformer blocks. ViT [3] was the first to introduce Transformer into the field of CV and achieved the best results in the image classification task of the year. Transformers were then applied to various fields of CV and achieved excellent results comparable to CNN.

In this paper, we consider the lane detection task as a prior based on weak symmetry and little change in relative position, and then proposed Cyclic_accumulation-Transformer(CaT) which introduced the transformer mechanism into CNN (combined the idea of RESA [3] or SCNN [18]). The overall architecture of our network refers to the Unet network [26] and is divided into three main parts: Encoder, CaT, and Decoder. The input image (assuming a 3-channel lane picture) goes through Encoder part (ResNet_based backbone, lightweight

backbone or another improved backbone) to extract high-level semantic information, and then the CaT module's feature_map can learn the relative relationship between pixels and pixels in a long distance range. At this point, CaT introduces a large amount of redundant information, but in the subsequent decoder module, this redundant information will be weakened, and only information similar to the label will be retained gradually.

In Unet networks [26] or other classical image segmentation algorithms [12], [16], [26], skip-connection is considered as an indispensable part, while in our proposed network structure, skip-connection can be removed without affecting the performance. In addition, the multi-head self-attention mechanism in the transformer structure can be optimized to a single-head self-attention mechanism in our CaT model.

Using the combination of transformer and CNN, our network can achieve a good performance with few training cycles on both the Tusimple and CULane without extra datasets (96.97% accuracy on Tusimple and 75.9% F1-score on CULane).

The main contributions can be summarized as follows:

- We propose the CaT module to obtain and integrate global information more efficiently; The CaT module is plug-and-play and can be easily integrated into other networks.
- In the new design of CaT, the "multiple heads" of the multi-headed attention mechanism can be removed and the final performance is not affected; The skip-connection in the classical segmentation network may not be applicable in the lane detection, as we have experimentally demonstrated.
- Our work is the first to combine Transformer and CNN in the field of lane detection, and we get state-of-the-art performance on the Tusimple without extra datasets.

II. Related Work

A. Segmentation_based methods in lane detection

The segmentation-based approach is the most straightforward and simplest way to detect lane lines, but classical segmentation networks that perform better in the general fields often perform poorly on the lane detection [18], [29], [37].

To fix this problem, SCNN [18] further improves the information flow delivery based on the idea of Markov Random Field (MRF) or Conditional Random Field (CRF) [15]. The feature_map that passes through the backbone firstly iterates row-by-row in "top to bottom" and "bottom to top", followed by column-by-column loop iterations of "left-to-right", "right-to-left" to obtain global information and further enhance the information of the occluded part. RESA [3] further improves the information transmission method in SCNN [18], and changes the information transmission from "row-by-row" and "column-by-column" to a certain stride. The above segmentation networks are often doing multi-classification tasks, that is, the number of lane lines in a picture needs to be defined in advance. LaneAF [1] introduced the concept of "affinity fields" to detect any number of lane lines without prior definition . It allows feature_map to perform binary classification tasks and train affinity fields at the same time. In the final judgment, the trained affinity fields are used to cluster trained binary lane images (that is, multiple classification = binary classification + clustering after affinity), but this method is less effective at the intersection of lane lines. Another usage scenario is to use the segmention_based network as a secondary branch to enhance the performance of the main branch network [21].

B. Other methods in lane detection

Row-wise_based approachs [13], [21] grid the image , converting each pixel predicted by the Segmentation_based method into line-by-line prediction of grid position coordinates containing lane line pixels, and finally mapping these coordinates back to the original image to obtain the final lane line coordinates.

Parameter_based approachs [14], [32], treated lane lines as a curve defined by parameters and predicted the parameters directly. Subsequently, LSTR [14] introduced Transformer into parameter_based detection, further improving speed and accuracy.

Key-point_based methods [9], [13], [32] are motivated by the fact that lane lines are composed of points, and then predicted the key points of lane lines directly.

C. The fusion of Transformer and CNN in deep-learning

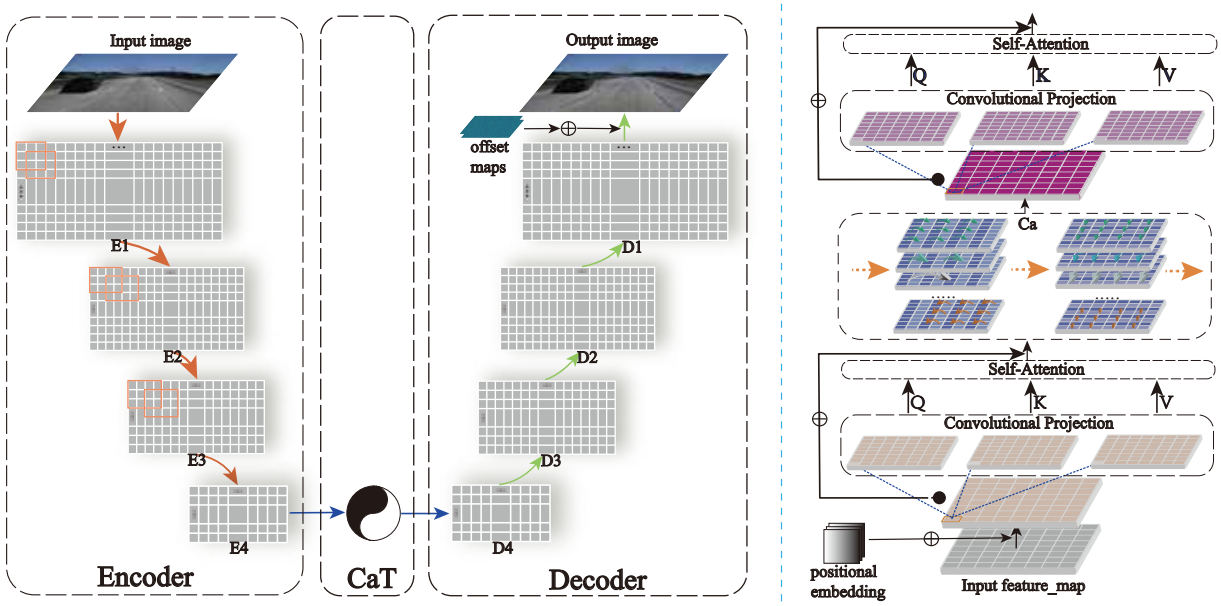
ViT [3] was the first effort by integrating the Transformer into CV and outperformed CNN in regards to classification results; ConViT [4] proposed a new SA Layer (Gated Positional Self-attention (GPSA)) to replace the SA layer in ViT, GPSA learns about gating parameters to determine whether to behave as a convolutional layer or not; CvT [35] proposed hierarchical multi-stage structure, each stage uses convolution to generate tokens and reduces computational costs by using different step sizes of convolution projection in the multi-head self-attention. Bottleck [28] outperformed the original ResNet in a variety of visual tasks simply by replacing the last three bottleneck layers in ResNet with the modified Transformer; Conformer [20] is a double parallel network: CNN Branch captures local feature and Transformer Branch captures global features, and the Feature Coupling Unit (FCU) is responsible for the information fusion of the two.

III. Methods

In this section, we illustrate the details of our proposed scheme, a novel combination of transformer and CNN patterns — CaT embedded in U-shape, a typical encoder-decoder segmentation network.

A. The overall structure

The overall network structure is shown in Fig. 1(a). The overall shape is U-Shape, assembled by three parts: Encoder, CaT and Decoder.



(a) Overall architectural (b) Overview of the CaT

Fig. 1. Details of the proposed method. (a) Encoder module uses standard ResNet to obtain multi-scale feature_maps (E1-E4). Decoder module uses bilinear interpolation and deconvolution to restore the intermediate feature_maps (D1-D4) to the original image size; Learnable offset maps are added one step ahead of the output. (b) The input feature_map added with positional embedding is mapped to Q, K, V by Convolutional projection module, followed by the calculation of self-attention, and then into the Ca model for information fusion, finally the Convolutional projection and self-attention modules are repeated.

1) Encoder: Encoder module is used to acquire high-level semantic information, which is divided into four convolution blocks. Given the matrix of image sizes (e.g., 3: RGB three channels; H: the length of the image; W: the width of the image), the image goes through four convolution layers and its length and width are halved respectively, and along with the increase of dimensions, the low-level semantic information is gradually transformed into high-level semantic information. Finally, feature_map of [C,H/8,W/8] is obtained.

2) CaT: CaT consists of two parts: Transformer and Cyclic_accumulation, which are used to integrate feature_map ([C,H/8,W/8]) with high-level semantic information obtained by Encoder modules. Transformer module includes position-embedding(PE) and self-attention(SA), firstly feature_map [C,H/8,W/8] plus position_embedding of the same size, then after convolutional_projection mapping into Q, K, and V of the same size. Q, K, and V calculate self-attention and input to Cyclic_accumulation module. The Cyclic_accumulation module first divides the input matrix into H rows, and then passes the information cumulatively from top to bottom and bottom to top; after that, it divides the input matrix into W columns, and then passes the information cumulatively from left to right and right to left. Finally, it passes a layer of self-attention module to further fuse the information from the previous two steps.

3) Decoder: The Decoder module is used to restore feature_map to its original size, along with a shift from high-level semantic information to low-level semantic information. Finally, a learnable offset_map is added to fine-tune the lane position of the network output. In the training phase, the feature_map output from the Decoder module calculates and minimizes the cross entropy loss with the given label during iterations; In the test phase, the output feature_map already shows the location of the specific lane lines and only needs to be overlaid on the original map as the final output.

B. Details of CaT

Our proposed CaT module is used to aggregate high-level semantic information, which is the core module of the entire network and consists of four main parts (as shown in Fig. 1(b)): Positional-embedding, Convolutional projection, Self-attention, and Cyclic_accumulation(Ca).The first three parts are basic elements for standard transformer-encoder, for Ca module, we follow the practices of SCNN [18] and RESA [3]. The combination of these four parts effectively solves the problem of "serious occlusion" in lane detection task, and enhances the imagination of detection of network.

Assuming that the input tensor is $X^{H,W,C}$, where H, W, and C represent rows, columns, and channels.

1) Position embedding.: Transformer was first applied in the field of NLP, and NLP sequence naturally has position attributes. When it introduced into the CV field [3], we need to carry out similar position embedding on the token (patch/ Pixel) of the image to add position information. Transformer's position embedding mainly includes two aspects: absolute position embedding and relative location embedding. We use learnable position embedding here and rely on input embedding as follows:

$$X^{H,W,C} = X^{H,W,C} + P^{H,W,C}, \tag{1}$$

Where $P^{H,W,C}$ indicates the added learnable position embedding has the same channels, rows, and columns as the input feature_map ($X^{H,W,C}$). $P^{H,W,C}$ initializes a gaussian random distribution with mean 0 and variance 1. It can be seen from the Fig. 4 ("after-pe") that feature_map added with position embedding looks like added Gaussian noise.

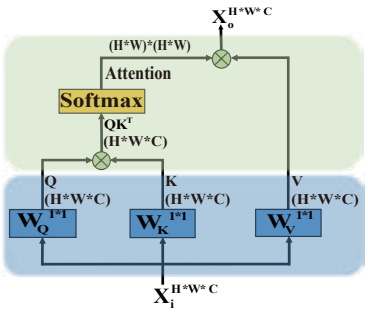


Fig. 2. Convolutional projection module (The green shaded area) and self-attention module (The blue shaded area).

2) Convolutional projection.: Convolutional projection maps tokens fused with location information into three different subspaces, namely Query, Key, and Value. Different from Linear projection in ViT, convolutional projection does not need to tensor 2D images into 1D as input (flatten operation). It's going to go straight through the trainable 1*1 convolution and then sliding through the tensor for 2D (As shown in blue part in Fig. 2). This computational progress can be formulated as:

$$X_{q/k/v}^{H,W,C} = Conv2d_{1*1}(X_i^{H,W,C}), \tag{2}$$

This operation has two advantages: one is that the computational cost is saved with fewer computational parameters; the other is that more spatial features can be retained compared with Linear Projection [3].

3) Self-Attention: The self-attention module is used to obtain the correlation (Attention matrix) within the data or features, and the vector product is calculated through the Q matrix and K matrix obtained by the convolution mapping. This kind of correlation can be global, and this parameter weight is dynamically changed with different input data. Finally, V matrix and Attention matrix are computed to produce globally dependent outputs. Our structure uses the self-attention module in two places, on either side of the Ca module (as shown in the green part of Fig. 2).The self-attention formulation is given as follows:

$$Attention = Softmax(\frac{(X_q^{H,W,C})(X_q^{H,W,C})^T}{\sqrt{d_k}}), \tag{3}$$

$$X_o^{H,W,C} = (Attention)(X_v^{H,W,C}), \tag{4}$$

4) Cycle_accumulation(Ca): We use the Ca module to realize the global information fusion method of the CNN mode. First, the feature_map tensor $X_i^{H,W,C}$ is divided into H rows and W columns by rows and columns, and then the information is accumulated cyclically in four directions: "top to bottom", "bottom to top", "left to right" and "right to left". The computation of Ca can be formulated as follows:

$$X_{i'}^{h,w,c} = X_i^{h,w,c} + f(\sum_{m=0}^{C}\sum_{n=0}^{W} X_i^{(h\pm s_k^H)modH,w+n,m}), \tag{5}$$

$$s_k^H = \frac{H}{2^{log_2 H-k}}, k = 0, 1, ..., log_2 H - 1, \tag{6}$$

$$X_{i'}^{h,w,c} = X_i^{h,w,c} + f(\sum_{m=0}^{C}\sum_{n=0}^{H} X_i^{h+n,(w\pm s_k^W)modW,m}), \tag{7}$$

$$s_k^W = \frac{W}{2^{log_2 W-k}}, k = 0, 1, ..., log_2 W - 1, \tag{8}$$

s_k^H represents the stride of the shift, which depends on the number of rows H and the number of columns W. f indicates the one-dimensional convolution layer. $'$ means the new formed tensor. Eq.5 and Eq.7 represent the cyclic accumulation process of different strides.

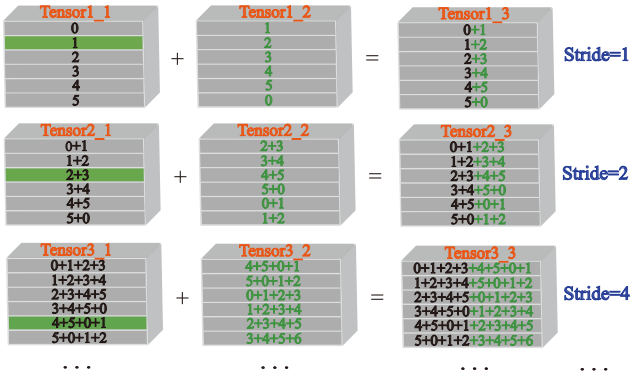


Fig. 3. Detailed information fusion process. Now, assume that the tensor has five rows, the first column representing the tensor to be processed, and the second column representing the new tensor formed after the specific stride. The number inside the tensor in the figure indicates the information contained in a certain line at this time.

Fig. 3 shows a more intuitive information flow transfer process, in which we take the "top to bottom" direction of information transfer as an example. In the formula in the first line(Stride=1), the left side of "+" represents the original "Tensor1_1" to be calculated, the right side of "+" represents the new "Tensor1_2" formed by the original "Tensor1_1" after stride=1, and the one on the right side of "=" is the "Tensor1_3" formed after information integration. It can be seen that each row in "Tensor1_3" has obtained the information of the next row. Similarly, in the second line(Stride=2), the initial tensor is the tensor from the previous step ("Tensor2_1" is the same as "Tensor1_3"), "Tensor2_2" is the new of "Tensor2_1" after the stride=2, each row in the resulting "Tensor2_3" contains 4 rows of information. Similarly, each line in "Tensor3_3" can collects 8 lines of information after stride=3.

In addition to the collection of global information in Ca mode, corresponding redundant information is also added. For

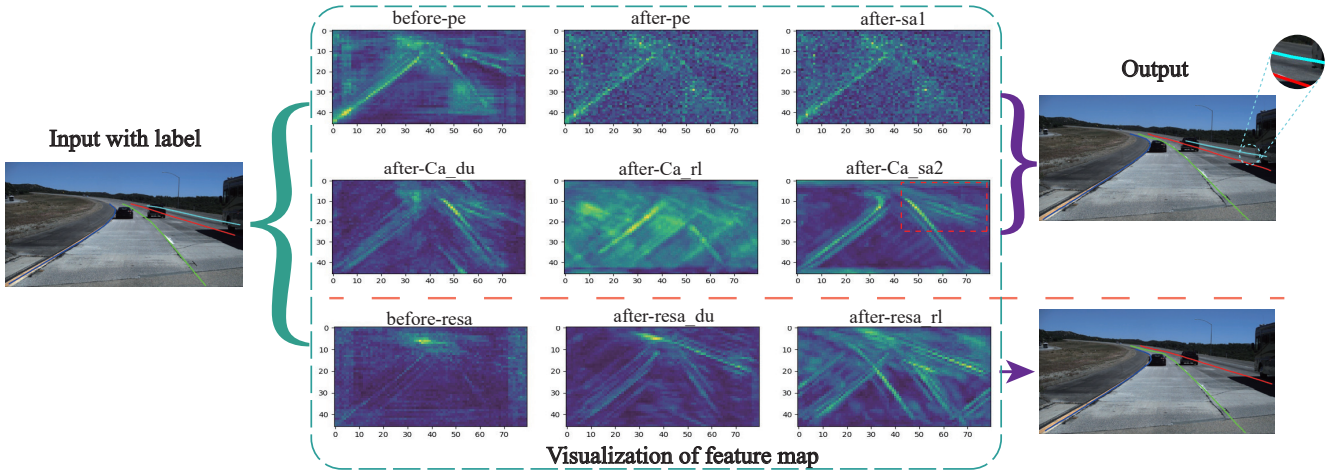


Fig. 4. Visualization of CaT (above the red dotted line) and RESA [37] (underneath the red dotted line) modules. The input of both models is the same image, and the label is represented by different colored lines in the picture.

example, information 0 and 1 in the first line of "Tensor3_3" in Fig. 3 are redundant information. Such aggregated redundant information is finally refined into high-level semantic information that can reflect lane location through the self-attention module (see "after-ca_sa2" in Fig. 4). Feature_map output from the earlier Ca module with Transformer is more chaotic than RESA (see Fig. 4 for a comparison of feature_map "after-ca_rl" and "after-resa_rl").

IV. Experiment

A. Datasets and evaluation metrics

1) Datasets: In order to verify the performance of our method, we adopt two most authoritative public datasets in lane detection - Tusimple and CULane.

Tusimple is a relatively old public dataset with images collected from highways. The dataset consists of about 7,000 one-second-long video clips of 20 frames each, of which the training set has 3626 video clips, 3626 annotated frames, and the test set has 2782 video clips. The resolution of each image is 720*1280 (720 means the height of the image, 1280 means the width of the image).

CULane is a publicly available dataset from [18], which is much larger and has more complex scenarios compared to the Tusimple. CULane contains approximately 120,000 images, of which 88,880 are in the training set and 34,680 in the test set; The images were grouped into 9 scenarios: "Normal"; "Crowded"; "Night"; "No line"; "Shadow"; "Arrow"; "Dazzle light"; "Curve"; "Crossroad". The resolution of each image is 1640*590 (590 means the height of the image, 1640 means the width of the image).

2) Evaluation Metrics: For Tusimple, three metrics are developed officially: Accuracy; FPR(False-Positive Rate); FNR(False-Negative Rate). The accuracy is calculated as follow:

$$Accuracy = \frac{\sum_{clip} P_{clip}^{pred}}{\sum_{clip} G_{clip}^{gt}}, \tag{9}$$

Where in each clip, P_{clip}^{pred} is the number of correctly lane points that have been predicted by the method (Compared with ground truth, the distance within a certain range is judged to be correct) and G_{clip}^{gt} is total number of ground truth points. At the same time, lane lines with accuracy greater than 85% are considered true positives (TP), otherwise they are considered false positives or false negatives.

For CULane, we refer to the unified calculation method F1-score given by [18] to judge our method. The calculation method of F1-score is as follows:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}, \tag{10}$$

Where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$ the calculation of TP, FP and FN are all based on the IoU (Intersection over Union) calculated by the predicted lane lines and Ground Truth label.

B. Implementation Details

In the data processing part: For Tusimple, we first resize the input image to 368*640 (height * width) as the input of the network. Similarly, for CULane, we resize the input image to 288*800 (height * width). In particular, since Tusimple itself does not provide segmentation annotations, we also need to use the supplied JSON file to generate segmentation annotations. In terms of data enhancement, we adopted the current commonly used strategies:ColorJitter, RandomResize, RandomCrop, RandomRotation and GroupNormalize.

In the training phase: we tried four pre-trained weight models (ResNet18, ResNet34, ResNet50 ResNet101) as backbone. We trained 400, 36 epochs for Tusimple and CULane respectively and finally obtain the best weights. Batch_size are set to 12. For the optimizers, SGD optimizer is adopted, in which the maximum learning rate parameter is set to 0.02, weight_decay is set to 1E-4, momentum is set to 0.9. Further, "LambdaLR" scheduler was used to dynamically adjust the learning rate. For the loss function, we use "Cross_entropy" and "Dice_loss" for Tusimple and CULane respectively.

All our models are trained with NVIDIA GeForce RTX 3090(24G) GPU or Tesla V100 PCIE(32G) GPU based on the Pytorch-1.10.0.

C. Result

Our method blends the advantages of Transformer and CNN in a feasible way, and receives good performance on both Tusimple and CULane. Our comparison method comes from the real-time ranking of Tusimple and CULane from the statistics above of: https://paperswithcode.com/sota, (especially FOLOLane [22], LaneAF [1], CondLane [13] are the latest papers just published in 2021).

Table I shows the comparison between our method and the state-of-the-art approaches in recent years on Tusimple. Our method with ResNet34 yields a new start-of-the-art accuracy of 96.97% with 47 FPS, while the FN is also much smaller than the other methods. It can be seen from Fig. 5 that our method can achieve good results in some places with serious occlusion and where the end lane lines will converge. In particular, our method can also achieve good generalization in places where the ground-truth label is slightly wrong.

TABLE I
COMPARISON AND STATE-OF-THE-ART RESULTS ON TUSIMPLE.

Methods	Accuracy	FP	FN
PolyLaneNet [32]	93.36%	0.0942	0.0933
LaneNet [17]	93.38%	0.0780	0.0224
CondLaneNet(ResNet34) [13]	95.37%	0.0220	0.0382
CondLaneNet(ResNet18) [13]	95.48%	0.0218	0.0380
LaneATT(ResNet18) [31]	95.57%	0.0356	0.0301
LaneAF(DLA-34) [1]	95.62%	0.0280	0.0418
LaneATT(ResNet34) [31]	95.63%	0.0353	0.0292
UFAST(ResNet18) [21]	95.82%	0.1905	0.0392
UFAST(ResNet34) [21]	95.86%	**0.1891**	0.0375
LaneATT(ResNet122) [31]	96.10%	0.0564	0.0217
LSTR [14]	96.18%	0.0291	0.0338
SCNN [18]	96.53%	0.0617	0.0180
CondLaneNet(ResNet101) [13]	96.54%	0.0201	0.0350
ENet-SAD [7]	96.64%	0.0602	0.0205
RESA(ResNet18) [37]	96.70%	0.0395	0.0283
RESA(ResNet34) [37]	96.82%	0.0363	0.0248
FOLOLane [22]	96.92%	0.0447	0.0228
CaT (ResNet34)	**96.97%**	0.0654	**0.0178**

For CULane, Table II shows the comparison between the results of our CaT method and the results of some other state-of-the-art methods, where our method obtained 75.9% of the F1score. Furthermore, our method performs best in the three scenarios with the highest proportion (Normal(27.7%), Crowded(23.4%) and Night(20.3%)). In addition, the FPR(False Positive Rate) value of Crossroad(9.0%) was also the lowest(1097).

D. Ablation study of CaT

In order to see the specific role of the Transformer more clearly, we split the components of the Transformer into PE (Position Embedding) and SA (Self-Attention) separately , and therefore discuss their specific performance in CaT separately, the results of CaT's ablation experiments are referred to Table III. The first line in Table III shows the result of baseline.

TABLE II
COMPARISON AND STATE-OF-THE-ART RESULTS ON CULANE. ONLY CROSSROAD IN THE TABLE SHOWS ITS PERFORMANCE IN FP.

Category (proportion)	ENet-SAD [7]	SCNN [18]	UFAST [21]	ERFNet-E2E [25]	PINet [9]	RESA [37]	**CaT (ours)**
Normal (27.7%)	90.1	90.6	90.7	91.0	90.3	92.1	**92.7**
Crowded (23.4%)	68.8	69.7	70.2	73.1	72.3	73.1	**74.0**
Night (20.3%)	66.0	66.1	66.7	67.9	67.7	69.9	**70.5**
No line (11.7%)	41.6	43.4	44.4	46.6	**49.8**	47.7	47.6
Shadow (2.7%)	65.9	66.9	69.3	74.1	68.4	**72.8**	69.6
Arrow (2.6%)	84.0	84.1	85.7	**85.8**	83.7	88.3	88.4
Dazzle light(1.4%)	60.2	58.5	59.5	64.5	66.3	**69.2**	64.7
Curve (1.2%)	65.7	64.4	69.5	**71.9**	65.6	70.3	68.8
Crossroad(9.0%)	1998	1990	2037	2022	1427	1503	**1097**
Total	70.8	71.6	72.3	74.0	74.4	75.3	**75.9**

The last line presents our final CaT scheme: PE+SA (before) +Ca+SA (after), they are all in series.

TABLE III
THE ABLATION EXPERIMENTS OF CAT ON TUSIMPLE. THESE MODULES ARE ALL CONNECTED IN SERIES.

Baseline	PE	SA(before)	Ca	SA(after)	Accuracy
✓					96.33
✓	✓	✓			96.79(+0.46)
✓			✓		96.85(+0.52)
✓		✓	✓		96.91(+0.58)
✓	✓		✓		96.90(+0.57)
✓	✓	✓	✓		96.91(+0.58)
✓			✓	✓	96.89(+0.56)
✓		✓	✓	✓	96.93(+0.60)
✓	✓	✓	✓	✓	96.97(+0.64)

It can be seen from the comparison results of the first three lines that Encoder-Decoder with the high-level semantic information fusion module has significantly higher performance in the framework of semantic segmentation than that without it, which indicates that the further global fusion of high-level semantic information is crucial for lane detection based on segmentation; In comparison with the third, fourth, fifth and sixth lines, it can be seen that the global information integration of Transformer and CNN can be well complementing their performance mutually and the performance is further improved as a result. As can be seen from Fig. 4, the Ca module with PE or SA is more chaotic in its high-level semantic information and contains obscured information; By comparing the third and seventh lines and the fourth and seventh lines, we can see the CA+SA (after) model to do further fusion of the chaotic information after the Ca module fusion (culling out the information needed by the Decoder module later) would be more desirable than letting the chaotic information go directly to the Decoder module, and also verifies the validity of SA (after). Finally, comparing the last row with all the above results, we verify the Transfomer and CNN can work together with better performance and they are more robust with this fusion method of CaT.

E. Extra experiments

1) Removing multi-heads in Transformer: The multi-head attention mechanism is generally considered where each head can compute and capture different feature subspaces, and this

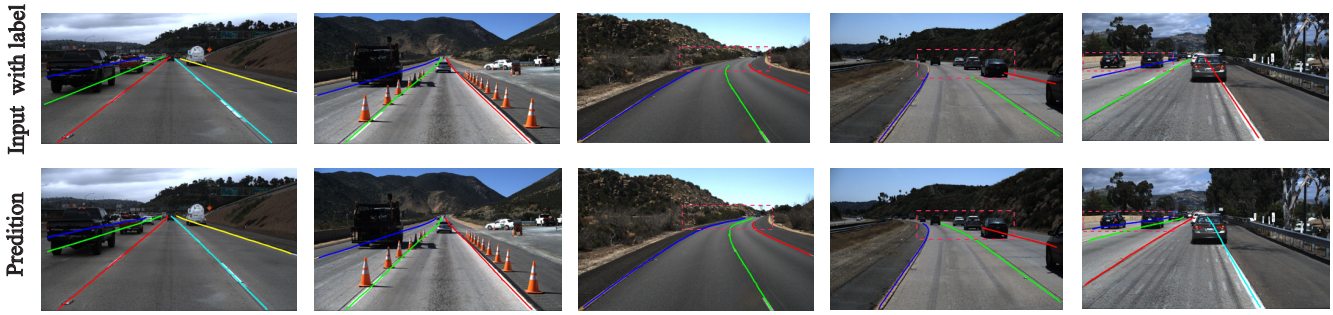


Fig. 5. Visualization results on Tusimple. Different lane lines are colored differently. The red box in the last three columns represents the part where the prediction is better than that of label.

multi-head structure is the default settings of many works using Transformer structures in the CV domain. Here, we experimentally observed whether this multi-head mechanism is also applicable to our CaT model. The result (Table IV) demonstrates that CaT model doesn't need the current mainstream "multi-head", and only using "single-head" is the best choice.

TABLE IV
THE EXPERIMENTAL RESULTS OF "MULTI/ SINGLE HEAD"(♢), "WITH/ WITHOUT SKIP-CONNECTION"(♡) AND FOUR PARALLEL FUSION MODES(♣) ON TUSIMPLE.

Methods	Accuracy
CaT (with 1 head)♢	**96.97%**
CaT with 2 heads♢	96.81%
CaT with 4 heads♢	96.84%
CaT (without Skip-connection)♡	**96.97%**
CaT with Skip-connection♡	96.91%
Add directly♣	96.89%
Learnable α and β ♣	96.87%
Concat and FFM♣	96.84%
Concat and CBAM♣	96.91%

2) Removing skip-connection in U-shape structure: Skip-connection is common in full convolutional network [16] and their variants [8], [26], [38], which can compensate the information lost in the Encoder process during the Decoder process. We test CaT with skip-connection, and the results are shown in Table IV. It is the authors' belief that the reasons for the performance declining are as follows: The main difficulties of lane detection comes from the cultivation of the imagination (1. Ignore of occluding objects, 2. Imagination of the shape of occluded objects) of the network for the occlusion part, the imagination of this part can be obtained from the high-level semantic information obtained by gradual convolution, while the low-level detail information is more about the contour information of objects (including the contour of occluding objects). The network using skip-connection operation also transmits the detailed contour information of occluding objects, which could interfere with the accuracy of the final result. It is the authors' belief that skip-connection operation is desirable for tasks that do not require cultivating the network's imagination for the occluded object, however, the skip-connection operation is not necessary for tasks that are seriously occluded and require the network to imagine the occluded object (such as lane detection task).

3) Fusion of Transformer and CNN: We take the parallel mode as the means of fusing the Transformer and CNN together, because the parallel mode is popular in the development of Transformer and CNN fusion [19], [20]. In details, we tried four parallel ways of fusing Ca and Transformer modules: Add directly, Learnable α and β, Concat and FFM, Concat and CBAM (shown in Fig. 6).

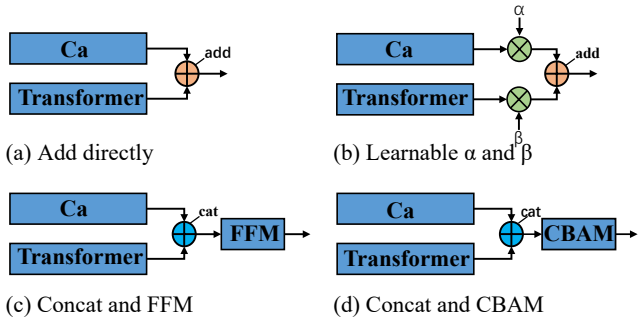


Fig. 6. The four different fusion methods in parallel mode. The FFM module is referenced from STDC network [5] and [34] for the CBAM module.

The results in Table IV show the performance of these four fusion modes respectively. The performance of parallel fusion mode is obviously higher than that of single Ca module and single Transformer module. But in comparison with the serial mode, parallel fusion mode has no absolute advantage.

CONCLUSION

In this paper, we propose a plug-and-play CaT module to solve the lane detection task, the essence of the work is to take advantages of the merits of CNN and Transformer respectively, making the network more expressive and imaginative. On the basis of CaT (a high-level semantic information fusion), our model achieves good results on the two popular datasets, e.g., CULane and Tusimple. Additional experiments further prove that Transformer and CNN can achieve better performance in a serial manner in our model, and CaT no longer needs multi-head self-attention mechanism and skip-connection structure due to the characteristics of lane detection task itself.

REFERENCES

[1] Abualsaud, H., Liu, S., Lu, D.B., Situ, K., Rangesh, A., Trivedi, M.M.: Laneaf: Robust multi-lane detection with affinity fields. IEEE Robotics and Automation Letters 6(4), 7477–7484 (2021).

[2] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).

[4] d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning. pp. 2286–2296. PMLR (2021).

[5] Fan, M., Lai, S., Huang, J., Wei, X., Chai, Z., Luo, J., Wei, X.: Rethinking bisenet for real-time semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9716–9725 (2021).

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016).

[7] Hou, Y., Ma, Z., Liu, C., Loy, C.C.: Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1013–1021 (2019).

[8] Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.W., Wu, J.: Unet 3+: A full-scale connected unet for medical image segmentation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1055–1059. IEEE (2020).

[9] 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).

[10] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012).

[11] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998).

[12] Lin, T.Y., Doll´ar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017).

[13] Liu, L., Chen, X., Zhu, S., Tan, P.: Condlanenet: a top-to-down lane detection framework based on conditional convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3773–3782 (2021).

[14] Liu, R., Yuan, Z., Liu, T., Xiong, Z.: End-to-end lane shape prediction with transformers. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 3694–3702 (2021).

[15] Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proceedings of the IEEE international conference on computer vision. pp. 1377–1385 (2015).

[16] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015).

[17] Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Towards end-to-end lane detection: an instance segmentation approach. In: 2018 IEEE intelligent vehicles symposium (IV). pp. 286–291. IEEE (2018).

[18] Pan, X., Shi, J., Luo, P., Wang, X., Tang, X.: Spatial as deep: Spatial cnn for traffic scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018).

[19] Pan, X., Ge, C., Lu, R., Song, S., Chen, G., Huang, Z., Huang, G.: On the integration of self-attention and convolution. arXiv preprint arXiv:2111.14556 (2021).

[20] Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., Ye, Q.: Conformer: Local features coupling global representations for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 367–376 (2021).

[21] Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection. In: European Conference on Computer Vision. pp. 276–291. Springer (2020).

[22] Qu, Z., Jin, H., Zhou, Y., Yang, Z., Zhang, W.: Focus on local: Detecting lane marker from bottom up via key point. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14122–14130 (2021).

[23] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018).

[24] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog 1(8), 9 (2019).

[25] Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems 19(1), 263–272 (2017).

[26] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015).

[27] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

[28] Srinivas, A., Lin, T.Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A.: Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16519–16529 (2021).

[29] Su, J., Chen, C., Zhang, K., Luo, J., Wei, X., Wei, X.: Structure guided lane detection. arXiv preprint arXiv:2105.05403 (2021).

[30] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015).

[31] Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Keep your eyes on the lane: Real-time attention-guided lane detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 294–302 (2021).

[32] Tabelini, L., Berriel, R., Paixao, T.M., Badue, C., De Souza, A.F., Oliveira-Santos, T.: Polylanenet: Lane estimation via deep polynomial regression. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 6150–6156. IEEE (2021).

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017).

[34] Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018).

[35] Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22–31 (2021).

[36] Yu, B., Jain, A.K.: Lane boundary detection using a multiresolution hough transform. In: Proceedings of International Conference on Image Processing. vol. 2, pp. 748–751. IEEE (1997).

[37] Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., Cai, D.: Resa: Recurrent feature-shift aggregator for lane detection. arXiv preprint arXiv:2008.13719 5(7) (2020).

[38] Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE transactions on medical imaging 39(6), 1856–1867 (2019).