Efficient Attention Calibration Network for Real-Time Semantic Segmentation

Hengfeng Zha[1] zhahengfeng@163.com
Rui Liu[1] * liurui@dlu.edu.cn
Dongsheng Zhou[1] * donyson@126.com
Xin Yang[2] xinyang@dlut.edu.cn
Qiang Zhang[1,2] zhangq@dlut.edu.cn
Xiaopeng Wei[2] xpwei@dlut.edu.cn
[1] *Key Laboratory of Advanced Design and Intelligent Computing (Ministry of Education), School of Software Engineering, Dalian University, Dalian, China, 116622*
[2] *School of Computer Science and Technology, Dalian University of Technology, Dalian, China, 116024*

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

In recent years, the attention mechanism has been widely used in computer vision. Semantic segmentation, as one of the fundamental tasks of computer vision, has been subject to tremendous development as a result. But because of its huge computing overhead, attention-based approaches are difficult to use for real-time applications such as self-driving. In this paper, we propose a self-calibration method baesd on self-attentiion that successfully applies the attention mechanism to real-time semantic segmentation. Specifically, a spatial attention module to adjust the edges of the coarse segmentation results which gained from the real-time semantic segmentation backbone network, and obtain more granular segmentation results. We refer to this method as the Efficient Attentional Calibration Network (EACNet). Experiments on the Cityscapes dataset validate the efficiency and performance of the method. With the high-resolution input and without any post-processing, EACNet achieved 72.4% mIoU of accuracy while running at 116.9 FPS. Compared to other state-of-the-art methods for real-time semantic segmentation, our network gained a better balance between performance and speed.

Keywords: Real-time Semantic Segmentation, Attention Mechanism, Computer Vision

1. Introduction

Image semantic segmentation is one of the fundamental tasks of computer vision and has a wide range of applications in areas such as autopilot and medical image diagnosis. Its purpose is to predict the category of each pixel in an image, i.e., pixel-level image classification. In recent years, a large number of high-accuracy semantic segmentation algorithms have been proposed, resulting in a significant improvement in the accuracy of the algorithms on major benchmark data sets. A large number of researchers set out to

* The second and third authors are Corresponding Authors.

enhance the real-time nature of semantic segmentation algorithms, giving birth to a range of real-time methods.

Current real-time semantic segmentation methods mostly use an encoder-decoder structure (Romera et al. (2018); Mehta et al. (2018)) based on a fully convolutional network (Shelhamer et al. (2017)). The structure uses a series of lightweight full convolution networks as encoders for the network and feeds the feature map extracted by the encoder into a lightweight decoder module to obtain segmentation results. Such a structure, while seemingly compact, actually uses lightweight decoders that are too simple to recover useful information from the feature map for segmentation to speed up the inference of the network. On the other hand, the encoder-only network (Wu et al. (2018)) proposed by some researchers further accelerates the network by abandoning the use of decoders and directly up-sampling the results obtained by the encoder network to obtain the final segmentation results.

While these networks can run in a real-time environment, their overly speed-seeking design makes it difficult to accurately reconstruct the spatial details that are lost during encoding. The lightweight decoder design doesn't even apply to the decoder making the partitioning results too coarse and is especially noticeable in terms of object edges and small object partitions. As a result, some works propose a multi-branch structure. Feature maps of different resolutions are extracted by using multiple lightweight networks in parallel, and these feature maps are then aggregated to obtain richer spatial information. But this approach takes too much advantage of the characteristics of the layers, and the network structure is too complex, with a lot of information and computational redundancy. And its performance is dependent on the lightweight network used.

The attentional mechanism enables the network to learn richer contextual information by establishing correlations between pixels. Some recent works have demonstrated that spatial attention has a significant effect on optimizing the segmentation of object edges, and they have designed an attention-based approach that significantly improves the accuracy of segmentation. But because of their huge computational volume, these spatial attention-based approaches are difficult to achieve real-time segmentation.

We propose to use spatial attentional mechanisms (Wang et al. (2018)) to optimize real-time semantic segmentation. To enable the attention mechanism to be used for real-time semantic segmentation, we propose a novel network structure called an efficient attention calibration network. Specifically, we propose a self-calibrating spatial attention embedding approach and design a self-calibrating attention module. A classification layer is used on the feature map to obtain the coarse segmentation results, which are then fed into the self-calibration attention module to obtain the final fine segmentation results. We establish correlations by treating coarse segmentation results rather than individual channels of the feature map, which is more conducive to minimizing gaps within classes while maximizing gaps between classes. And this approach avoids processing large amounts of channel information and reduces computational redundancy. As far as we know, there are not many ways to introduce attentional mechanisms into real-time semantic segmentation and to achieve a fast and effective framework for real-time semantic segmentation.

In summary, our main contributions can be summarized as follows:

- We propose a novel attentional method called Self-Calibration Attention Module for real-time semantic segmentation, which establishes a correlation for each object to be segmented.
- Based on the Self-Calibration Attention Module, we propose Efficient Attention Calibration Network(EACNet), which uses the attention mechanism for real-time semantic segmentation.
- Experiments on the cityscapes dataset demonstrate that EACNet can improve the performance of existing high speed semantic segmentation methods, especially at object edges.

2. Related Work

Convolutional neural networks were originally created to solve the image classification task (Deng et al. (2009)). FCN (Shelhamer et al. (2017)) is a pioneer of convolutional neural networks for semantic segmentation tasks. It removes the fully connected layer of VGG-16 (Simonyan and Zisserman (2015)) and replaces it with a convolutional layer, achieving pixel-level classification of images. Subsequent studies based on FCN have produced a large number of variants and improvements. As a result, the accuracy of image semantic segmentation has improved considerably.

Real-time semantic segmentation

ENet (Paszke et al. (2016)) is arguably the first attempt at real-time semantic segmentation. It achieves high-speed inference by reducing the number of network channels. However, this increase in speed leads to a significant decrease in accuracy. To solve this problem, Romera et al. (2018) designed a new encoder-only network by using residual connections (He et al. (2016)) and convolutional decomposition (Chollet (2017)) to construct a new convolutional block. Accuracy is ensured while efficiency is maintained. The authors of CGNet Wu et al. (2018) propose a Context Guided module for learning local features and features of the surrounding environment. The module consists of a regular convolutional nucleus and an expanding convolution, which form a parallel structure. They propose CGNet, which builds a decoders lightweight FCN network that achieves better real-time accuracy while maintaining network accuracy. Li and Kim (2019) proposes a more efficient Depth-wise Asymmetric Bottleneck (DAB) module that improves the CG module using depth separation convolution and convolutional decomposition, and again compresses the depth of the network with faster operation and higher accuracy than CGNet. Lo et al. (2019) uses an asymmetric convolutional structure with a combination of expanded convolution and dense connections to achieve high efficiency with low computational cost and model scale.

On the other hand, some multi-branching approaches are proposed. Zhao et al. (2018a) proposed ICNet and Oršic et al. (2019) designed SwiftNet use multiple resolution inputs of images to build multi-branch networks that take full advantage of the semantic information of low-resolution plots and the detailed information of high-resolution plots to enable the network to recover and refine segmentation predictions step-by-step with low computational

effort. Yu et al. (2018) proposed BiSeNet which use a two-path network to take into account both accuracy and speed. Li et al. (2019a) designed sub-network aggregation and sub-stage aggregation to aggregate features of different stages of the three backbone networks to improve the network speed while maintaining segmentation accuracy. These multi-branching approaches are mostly based on a fast and efficient backbone network, so we believe that the multi-branching approaches rely on single-branch network performance improvements.

Attentional Models

Attentional mechanisms have been applied in various areas of deep learning in recent years and Vaswani et al. (2017) uses them for machine translation. In the field of computer vision, Hu et al. (2018) constructed a channel-level attention module by extracting channel-level information and propose Squeeze-and-Excitation Networks. On the other hand, Wang et al. (2018) proposed Non-local Neural Networks which use of spatial attention mechanisms to capture long-range dependencies between pixels. PSANet (Zhao et al. (2018b)) gains contextual information from all locations of the feature map by learning an adaptive attention map for each location connected to other locations. Fu et al. (2019) proposed DANet which use adaptive attention to capture contextual information. CCNet(Huang et al. (2019)) acquires contextual information from the surrounding pixels on the cross path of each pixel by building a cross attention module. While all of these methods have achieved significant improvements, their attentional modules are all computationally complex, greatly reducing the speed of the algorithm's reasoning. To improve the speed of the attentional model, Zhu et al. (2019) proposes ANNNet, which use asynchronous Non-local to reduce the computational resources and improve the performance of Non-local Networks. At the same time, Li et al. (2019b) proposed to express the attentional mechanism as an expectation-maximizing approach, and in doing so, made a more compact estimate of the attentional graph, which greatly reduced the use of memory and computational complexity.

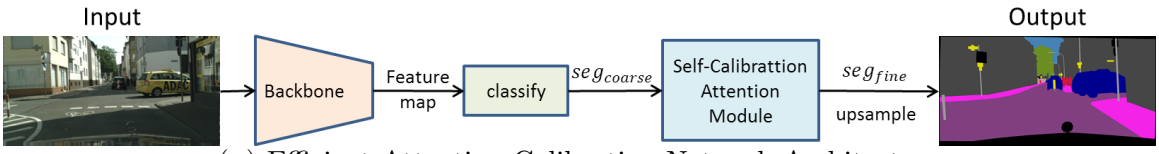
Inspired by DANet but different from it, we propose to use the attentional mechanism for the output of real-time semantic segmentation rather than feature maps, thus reducing the large consumption of computational resources by the attentional module. On the other hand, inspired by ANNNet(Zhu et al. (2019), they scaled the feature vector after point-wise convolution in Non-Local Network. We use the segmentation maps with its downsampled maps to solving the attention graph, this can further speed up the inference of all networks.

3. Methodology

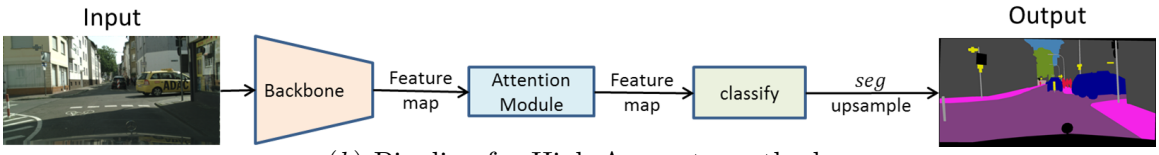
In this section,the architecture of the proposed network EACNet is introduced (shown in Figure 1(a)subfigure). The self-calibrating attention module used in our method is then shown. Finally, we describe how to build EACNet to complete real-time semantic segmentation in conjunction with the current real-time semantic segmentation network.

3.1. Network Architecture

We found that the self-attention module was applied to learn long-term dependencies between classes and improve compactness and semantic consistency within classes. The



(a) Efficient Attention Calibration Network Architecture



(b) Pipeline for High-Accurate methods

Figure 1: Pipeline Comparison of EACNet and High-Accuracy Attention Methods

previous approachs (Wang et al. (2018); Fu et al. (2019)) was to use a feature map F input module with a channel C (the network structure is shown in Figure 1(*b*)subfigure).

$$Seg = \mathcal{C}la(Attention(\mathcal{F})) \tag{1}$$

where $\mathcal{F}$ represents the feature map extracted by the backbone network, $Attention(\cdot)$ represents the whole attention module operation, $\mathcal{C}la(\cdot)$ represents the classification layer and Seg represents the segmentation result.

A notable problem is that the amount of operations is too large, and such operations create a lot of semantic redundancy at the channel level. Since the purpose of the attention module is to learn intra-class and inter-class correlations, the number of classes that are also used for real-time semantic segmentation on autopilot is usually limited (Cityscapes has 19 classes). Therefore, we propose an attentional calibration embedding method. The method uses feature maps extracted from the backbone network to obtain coarse segmentation results Seg_{coarse}(whose channel is equal to the kinds of segmentation target) through the classification layer. Then, use Seg_{coarse} as an input to the attention module to establish correlations directly to the individual pixel results of the segmentation map, thus achieving a refinement of the coarse segmentation result:

$$Seg_{fine} = Attention(Seg_{coarse}) \tag{2}$$

the coarse segmentation result Seg_{coarse} can be expressed as:

$$Seg_{coarse} = \mathcal{C}la(\mathcal{F}) \tag{3}$$

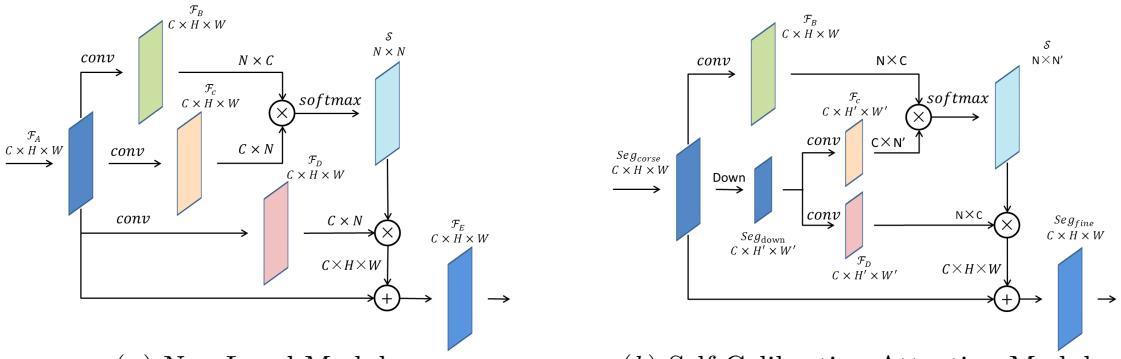
Combined with equation (2) and (3), the entire network pipeline can be expressed as:

$$Seg_{fine} = Attention(\mathcal{C}la(\mathcal{F})) \tag{4}$$

Following this line of thought, we propose a method called the Attentional Calibration Network. Figure 2(*b*)subfigure represents the structure of our network. Compared to the structure of Non-Local(Wang et al. (2018))(shown in Figure 2(*a*)subfigure), the approach

we use makes the network more compact and more consistent with attention in its sense. Specifically, for each of the categories to be classified, we establish correlations within their classes. Such a network model is more like a refinement of the segmentation results of the network using the self-attentive mechanism, which can be seen as an internal calibration. At the same time, such a structure simplifies the amount of computation and computational overhead and improves the efficiency of the network. To differentiate the attention modules used by other networks,the attention module used in this paper is named as Self-Calibration Attention Module. The entire network architecture is called the Efficient Attention Calibration Network (EACNet).

3.2. Self-Calibration Attention Module



(a) Non-Local Module (b) Self-Calibration Attention Module

Figure 2: Self-Attention Module

First, consider the Non-local module used in method (Wang et al. (2018)) (shown in Figure 2(a)subfigure), where the feature map $\mathcal{F}_A \in \mathbb{R}^{C\times H\times W}$ extracted from the backbone network is used as input to the module, and three new feature maps $\mathcal{F}_B$,$\mathcal{F}_C$,$\mathcal{F}_D$ are generated by the corresponding convolution, where $\mathcal{F}_B$,$\mathcal{F}_C$,$\mathcal{F}_D \in \mathbb{R}^{C\times W\times H}$. Then reshape them to $\mathbb{R}^{C\times N}$, where $N = H \times W$. then the $\mathcal{F}_B$ is transposed and multiplied with $\mathcal{F}_C$, and apply the softmax function to the matrix multiplication result to obtain the spatial attention graph $S \in \mathbb{R}^{\mathbb{N}\times\mathbb{N}}$.

$$S = softmax(\mathcal{F}'_B, \mathcal{F}_C) \tag{5}$$

Each value in S indicates the degree of similarity between the two locations in the feature map, and the more similar the feature representations of the two locations, the stronger the correlation between them (Li and Kim (2019)). Then, the resulting attention map S was multiplied with the feature map $\mathcal{F}_D$ by another matrix multiplication, and the resulting results were reconstructed as $\mathbb{R}^{C\times W\times H}$. Finally, multiplying this result by a scale parameter α performs element-level addition with the module's input $\mathcal{F}_A$ to obtain the module's final output $\mathcal{F}_E \in \mathbb{R}^{C\times H\times W}$:

$$\mathcal{F}_E = \alpha(S \cdot \mathcal{F}_\mathcal{D}) + \mathcal{F}_A \tag{6}$$

where α is a weight initialized to zero, which constantly changes itself during network training. Combined with equation (5) and (6), the entire attention module can be expressed

as:

$$\begin{aligned} Attention(\mathcal{F}_A) &= \alpha(S \cdot \mathcal{F}_{\mathcal{D}}) + \mathcal{F}_A \\ &= \alpha(softmax(\mathcal{F}_{B'} \cdot \mathcal{F}_C) \cdot \mathcal{F}_D) + \mathcal{F}_A \\ &= \alpha(softmax(h'(F_A) \cdot g(\mathcal{F}_A)) \cdot f(\mathcal{F}_A)) + \mathcal{F}_A \end{aligned} \tag{7}$$

Throughout the attention module, similar semantic features achieve mutual gain, thus imporving intra-class compact and semantic consistency. It is known from Equation (7) that the final feature of each location is the weighted sum of the features of all locations with the original features. As a result, it has a global context view and selectively aggregates contexts based on the spatial attention map.

The direct use of attentional modules causes a depletion of computational resources, A customary method is reduce the number of channels of the feature map $\mathcal{F}_B$, $\mathcal{F}_C$, and $\mathcal{F}_D$ by point-wise convolution. Specifically, it will take the feature graph $\mathcal{F}_A \in \mathbb{R}^{C \times W \times H}$ to get the feature graph $\mathcal{F}_B, \mathcal{F}_C, \mathcal{F}_D \in \mathbb{R}^{C' \times W \times H}, C' = C/4$. However, since the input of the attention module of EACNet is the coarse-segmentation results, and the whole module is built on the segmentation maps of the individual categories, the reduction of channels will make the correlation of some categories not be established accurately. Therefore, We propose a simpler and more efficient way to designed the self-calibration module. Specifically, to make the module lighter and less time-consuming, we first downsample the input coarse-segmentation results:

$$Seg_{down} = Down(Seg_{coarse}) \tag{8}$$

Then, generate a spatial attention map $Seg_{fine} \in \mathbb{R}^{C \times N}$ based on the downsampling results Seg_{down}:

$$\begin{aligned} Seg_{fine} &= Attention(Seg_{coarse}) \\ &= \alpha(softmax(h(Seg_{coarse}) \cdot g(Seg_{down}) \cdot f(Seg_{down}))) + Seg_{coarse} \end{aligned} \tag{9}$$

Where $Down(\cdot)$ denotes the downsampling operation, and the resulting $Seg_{down} \in \mathbb{R}^{C \times N'}$, $N' = W' \times H'$ is the resolution of the downsampled coarse-segmentation result Seg_{fine}.

3.3. Attentional Calibration Embedding

As shown in Figure 1(*a*)subfigure, EACNet consists of a backbone network and an attentional calibration module. In theory, EACNet can use any regular single-branch network as the backbone. In this paper, we use the state-of-the-art single-branch real-time semantic segmentation network as the backbone for EACNet. Specifically, it can be divided into two cases. the first one is that our network can use any encoder of the encoder-decoder network as the backbone. The feature map extracted by the encoder is passed through the classifier to get the coarse segmentation results, which are then fed into the self-calibration module. In another case, we use the whole encoder-only real-time semantic segmentation network as the backbone for EACNet. the segmentation results are fed into the self-calibration module. The detailed will be described in the experimental section below.

4. Experiments

In this section, we will experimentally verify the validity of our approach. This chapter is divided as follows: the first section describes our experimental setup, including the data set, the platform for training evaluation, and the setup of hyperparameters. In the second section, a series of comparison experiments are set up to verify the enhancement of our method relative to baseline models. In the end, we compare our method with other state-of-the-art methods.

4.1. Experiment Settings

Datasets

Cityscapes (Cordts et al. (2016)) is an urban segmentation dataset.It contains 5,000 finely labeled images and 20,000 roughly labeled images collected from 50 different cities. Each image has a resolution of 1024 $\times$ 2048, with 19 classes for semantic segmentation evaluation. We train our network using only finely labeled images. They were divided into three sections: 2975 for the training, 500 for the validation, and 1525 for the test.

CamVid(Brostow et al. (2009)) is another street scene dataset which images are annotated into 11 classes. It contains 701 annotation frames are divided into 367,101 and 233 for train, validation and test respectively.

Network

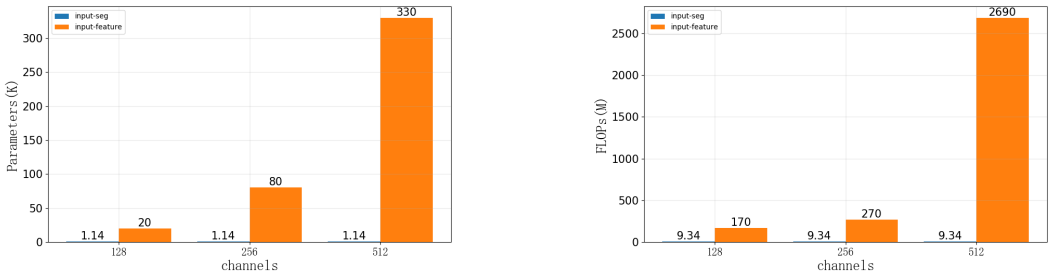
To demonstrate the performance of EACNet, we choose ERFNet (Romera et al. (2018)) and DABNet (Li and Kim (2019)) as the baseline models. ERFNet is a real-time semantic segmentation network with an encoder-decoder structure, and the DABNet is a encoder-only ResNet-like network. We use the encoder part of ERFNet and the entire network of DABNet as the backbone of EACNet, by removing the decoder structure of ERFNet and adding a classifier to its encoder to generate the coarse partition results, which are then fed into our proposed self-calibration module and generate the final partition results. For DABNet, we added the self-calibration module directly to the network's classifier and used it to get our results. Whether we use ERFNet or DABNet as our backbone, our network is trained in a hierarchical way, i.e., the backbone is trained first, then the trained backbone is used to initialize EACNet and train the entire network. The final experiment proves that our EACNet can enhance the performance of existing high-speed semantic segmented networks and that this performance does not slow down the reasoning of the network.

Implementation Details

We implement our method based on Pytorch. All evaluation metrics are done on a single RTX2080Ti graphics card, Ubuntu OS. Our hyperparameters are consistent with the original network settings. For ERFNet, however, we used the online difficult sample mining (OHEM) strategy (Shrivastava et al. (2016)) to train hard-to-optimize categories. For data enhancement, we used random horizontal flip, mean subtraction, and random scale transformations for the input images during training. Dimensions include 0.75,1.0,1.25,1.5,1.75,2.0. We trained by randomly cropping the image to a fixed size (1024x512). We do not use any

other data set for pre-training. No data enhancement techniques and testing techniques are used during testing.

4.2. Comparison Experiments



(*a*) Comparison of Parameters (*b*) Comparison of FLOPs

Figure 3: Effect of Different Inputs on Attention Module

Attentional Calibration Embedding is Efficient

To verify that EACNet's attention module uses fewer computational resources than the attention module of the high-accuracy method, Comparison of EACNet's attentional calibration embedding method with the attentional module of the high accuracy method in terms of parameters and FLOPs. The resolution of the experimental inputs was uniformly 64×128. As shown in Figure 3, since EACNet relies on the number of segmentation classes, it has a huge advantage in the cityscapes dataset of 19 target categories, both in terms of parameters and FLOPS, even the feature map with only 128 input channels (yet actually much more than that), our method has a nearly twenty-times advantage.

Experiments on Encoder-Decoder Baseline Model

Table 1: Comparison of Performance and Speed on Cityscapes Validation Set

Model	FLOPS	Parameters	Speed(ms)	FPS	Mean IoU(%)
ERFNet-base	30.055 B	2.067 M	11.76	85	72.46
ERFNet-enc	21.698 B	1.876 M	8.20	121.95	70.60
EACNet-erf-nl	21.707 B	1.878 M	10.26	97.46	73.94
EACNet-ERF	21.702 B	1.878 M	8.57	116.95	73.74

Comparative experiments in the Cityscapes validation set demonstrates EACNet's performance. Specifically, we reproduced ERFNet and compared several different variants: the full ERFNet(ERFNet-base), the ERFNet encoder with classify layers(ERFNet-enc), and EACNet using the ERFNet encoder as the backbone(EACNet-ERF). The results are shown in Table 1.

Compared to the results reported by the authors in their paper (70.0% mIoU in Cityscapes valid sets) (Romera et al. (2018)), our replicated ERFNet-base has better performance on validation sets (72.46% on mIoU), which may be due to our use of OHEM to train classes which are difficult to optimize. The ERFNet-enc only use the encoder is faster than the full ERFNet, but at the same time, the accurate has decrease. However, compared with ERFNet encoder, EACNet-ERF uses ERFNet encoder as the backbone improved 3.14% on mIoU. It also has a more than 1% mIoU improvement over the full ERFNet (ERFNet-base). More importantly, EACNet-ERF is more accurate and at the same time faster (3 ms faster, nearly 30% improvement) than the baseline method of ERFNet-base.

We also compared the performance of EACNet-ERF with the EACNet uses the original non-local module as the self-calibrating module(EACNet-erf-nl). The results(shown in Table 1) indicate that, compare to Non-Local module, the self-calibrating module allows faster inference speed(1.63 ms faster and nearly 15% faster) while maintaining the same accuracy(only 0.02% mIou of performance degradation). More importantly, the results on the cityscapes test sets show that our self-calibrated modules have better generalization capabilities (shown in Table 2).

Table 2: Comparison of performance and speed on cityscapes validation set

Method	Ave	road	side	buil	wall	fenc	pole	ligh	sign	vege	terr	sky	pers	rid	car	truc	bus	trai	moto	bic
ERFNet-base	72.5	97.8	82.8	91.6	53.5	58.6	63.0	65.5	74.7	91.9	62.3	93.5	79.0	57.5	93.7	63.3	74.6	44.5	55.7	73.5
EACNet-erf-nl(val)	73.9	97.8	82.8	91.2	58.9	59.3	61.1	64.6	73.2	91.5	61.2	92.9	79.1	56.5	93.6	63.4	80.4	67.9	56.3	73.2
EACNet-ERF(val)	73.7	97.7	82.6	91.4	57.5	56.8	60.9	65.0	74.1	91.7	62.3	93.5	79.0	56.0	94.1	70.8	79.1	59.9	55.4	73.4
EACNet-erf-nl(test)	71.0	97.9	81.9	90.9	52.6	51.2	59.3	66.3	70.8	92.3	70.7	94.3	82.3	63.2	94.1	53.0	56.9	46.1	56.1	68.5
EACNet-ERF(test)	72.4	98.0	82.4	91.0	51.3	52.8	59.3	65.9	70.9	92.4	71.1	94.0	82.0	63.1	94.2	57.1	67.2	56.8	56.9	69.5

On the other hand, we qualitatively analyzed the improvement of our approach over the baseline approach (shown in Figure 4). Compared to ERFNet, since our method is able to establish intra-class correlations between pixels, in the first graph (first row), relative to ERFNet (third column of the first row), EACNet (fourth column of the first row) eliminates the intraclass ambiguity of the car on the right side of the graph, and the car has a more refined profile. In the second and third images, it can be seen that EACNet has not only improved the edges of the objects, but also the identification of some small objects is more accurate (the rider in the middle of the second image and the head of the pedestrian obscured by vehicle on the right side of the third image).

Experiment on Encoder-only Baseline Model

As discussed in section 3, EACNet can be well embedded in any current single-branch real-time semantic segmentation framework. To verify that EACNet also works well for encoder-only networks. We reproduce DABNet and use its entire network as the encoder for EACNet. The output from DABNet's classification layer as input to our self-calibration module. The results are shown in Table 3. Compared to the baseline model, EACNet has a significantly improvement of 2.61% mIoU with only a small increase in parameters and calculations (0.001M for the parameter increase and 0.015B more for the FLOPs). In terms of inference speed, the reduction of 4 FPS seems insignificant compared to the speed of 101.94 FPS. At the same time, such a result also demonstrates that our approach can

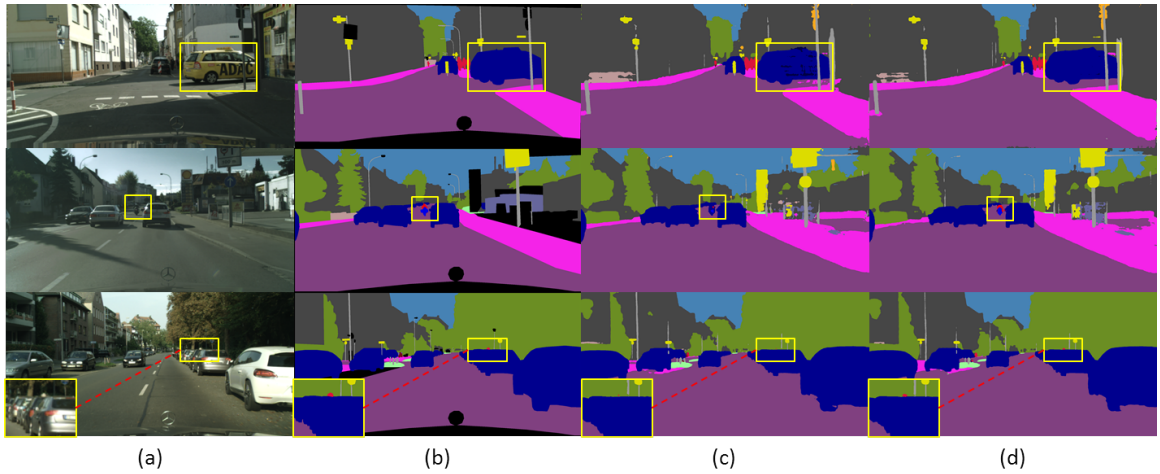


Figure 4: Visualization results on the Cityscapes validation dataset. From left to right are (a) Input, (b) Ground truth, (c) ERFNet and (d) EACNet

substantially improve the performance of encoder-only networks with only a few additional parameters and computations.

Table 3: Comparison of performance and speed of EACNet using DABNet as a backbone on the Cityscapes validation set.

Model	FLOPS	Parameters	Speed(ms)	FPS	Mean IoU(%)
DABNet	10.375 B	0.757 M	9.81	101.94	69.57
EACNet-DAB	10.380 B	0.758 M	10.21	97.92	72.18

4.3. Comparison with the state-of-the-arts methods

Through a series of experiments, our EACNet is able to bring performance improvements to current single-branch real-time semantic segmentation methods. It also improves the performance of the codec structure network by reducing the time consumed for reasoning and greatly increasing the reasoning speed. Finally, we compare EACNet to the current state-of-the-art single-branch model on the cityscapes test set. All results are derived from the performance of the methods reported individually on the paper or on the official cityscapes server. The final results of the comparison are shown in Table 4.

As can be seen from the results reported in Table 4, compared to other state-of-the-art methods, our model has a new performance on the trade-off between speed and performance. Compared to our baseline models ERFNet and DABNet, our approach has better performance on the test set (72.4% vs 68.0%, and 71.9% vs 70.1% on mIoU). And the advantages of our approach are even more pronounced for the instance-level metric miIoU

(Cordts et al. (2016)) for the key classes of scenarios. Compared to other approaches, our approach has significant advantages in both mIoU and miIoU. In terms of speed, RPNet (Chen et al. (2019)) is a bit faster than our approach (only 7 FPS), but its performance is far less than our approach (4.5% mIoU less). And it should be noted that the methods we report are based on single-scale tests and no testing tricks is used(including no TensorRT optimized inference).

The results of our method compared with other methods on per classes are shown in Table 5. It can be seen that EACNet exceeds the current state-of-the-art approach in almost every class. And the performance of the iIoU metric is far superior to other real-time methods.

We also evaluated our network on the CamVid dataset. The results as shown in the Table 6. our EACNet again achieves outstanding performance in efficiency and accuracy. The visualization results are shown in Figure 5.

Table 4: Comparison of performance and speed with other state-of-the-art methods on the Cityscapes test set.

Method	Pretrain	InputSize	mIoU(%)	miIoU(%)	FPS	Params	FLOPs
ENet	ImageNet	512×1024	58.3	34.36	76.9	0.37M	-
ERFNet	No	512×1024	68.0	40.42	41.7	2.07M	-
ESPNet	No	512×1024	60.3	31.82	112	-	4.5B
ESPNetv2	No	512×1024	66.2	36.03	-	-	2.7B
CGNet	No	1024×2048	64.8	35.89	50	0.5M	-
DABNet	No	512×1024	70.1	42.86	104.2	0.76M	-
EDANet	No	512×1024	67.3	41.78	108.7	0.68M	8.97B
RPNet(ERFNet)	No	512×1024	67.9	44.9	123	1.89M	20.71B
EACNet-DAB	No	512×1024	71.9	47.39	97.9	0.76M	10.38B
EACNet-ERF	No	512×1024	72.4	46.9	116.9	1.87M	21.70B

Table 5: Comparison of performance per classes on the cityscapes test set.

Method	roa	sid	bui	wal	fen	pol	lig	sig	veg	ter	sky	per	rid	car	tru	bus	tra	mot	bic	mIoU	miIoU
ENet	96.3	74.2	85.0	32.2	33.2	43.5	34.1	44.0	88.6	61.4	90.6	65.5	38.4	90.6	36.9	50.5	48.1	38.8	55.4	58.3	34.4
ERFNet	97.7	81.0	89.8	42.5	48.0	56.2	59.8	65.3	91.4	68.2	94.2	76.8	57.1	92.8	50.8	60.1	51.8	47.3	61.6	68.0	40.4
ESPNet	95.7	73.3	86.6	32.8	36.4	47.1	46.9	55.4	89.8	66.0	92.5	68.5	45.8	89.9	40.0	47.7	40.7	36.4	54.9	60.3	31.8
ESPNetv2	97.3	78.6	88.8	43.5	42.1	49.3	52.6	60.0	90.5	66.8	93.3	72.9	53.1	91.8	53.0	65.9	53.2	44.2	59.9	66.2	36.0
CGNet	95.9	73.9	89.9	43.9	46.0	52.9	55.9	63.8	91.7	68.3	94.1	76.7	54.2	91.3	41.3	56.0	32.8	41.1	60.9	64.8	35.9
DABNet	96.8	78.5	91.0	45.4	50.2	59.1	65.2	70.8	92.5	68.2	94.7	80.6	58.5	92.7	52.7	67.3	51.0	50.5	65.7	70.1	42.9
EDANet	97.8	80.6	89.5	42.0	46.0	52.3	59.8	65.0	91.4	68.7	93.6	75.7	54.3	92.4	40.9	58.7	56.0	50.4	64.0	67.3	41.8
RPNet	97.9	81.2	89.8	40.2	45.7	56.3	61.6	67.8	91.7	68.0	94.5	78.2	57.4	92.9	48.3	57.8	56.1	49.6	62.2	68.3	43.6
EACNet-DAB	98.0	82.6	91.3	49.4	51.9	58.9	66.3	71.3	92.5	70.5	94.6	81.8	61.4	94.0	54.5	68.0	53.7	54.8	69.6	71.9	47.4
EACNet-ERF	98.0	82.4	91.0	51.3	52.8	59.3	65.9	70.9	92.4	71.1	94.0	82.0	63.1	94.2	57.1	67.2	56.8	56.9	69.5	72.4	46.9

5. Conclusion

In this paper, a new single-branch real-time semantic segmentation structure is proposed in which the semantic segmentation results are entered into the attention module as feature graphs. Thus, the application of the self-attention approach to real-time semantic

Table 6: Comparison of performance and speed with on the CamVid test set.

Method	mIoU(%)	FPS	Params	FLOPs
ENet	51.3	111	0.37M	1.50B
ERFNet	65.0	133	2.07M	8.43B
ESPNet	62.6	205	0.68M	0.87B
CGNet	65.6	-	0.5M	-
DABNet	66.4	104.2	0.76M	-
EDANet	66.4	-	0.68M	8.97B
RPNet(ERFNet)	64.82	149	1.89M	6.78B
EACNet-DAB	69.6	100	0.76M	3.42B
EACNet-ERF	69.3	123	1.87M	7.15B

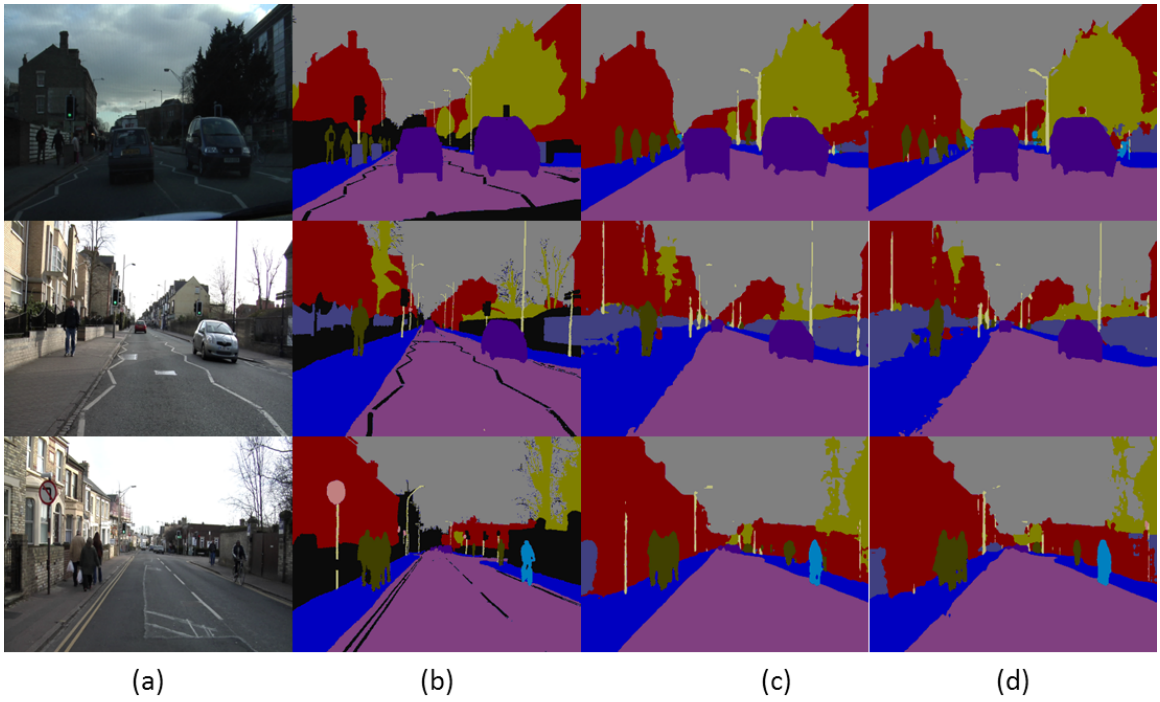


Figure 5: Visualization results on the CamVid (480×360) test dataset. From left to right are (a) Input, (b) Ground truth, (c) EACNet-ERF and (d) EACNet-DAB

segmentation networks is achieved. And the advanced nature of this structure has been experimentally demonstrated. At the same time, we have optimized the self-attention module and proposed an attention module that is more suitable for our attention calibration network. More accurate and efficient segmentation is achieved. Experiments on benchmark dataset demonstrate the pervasiveness of our approach to the current single-branch real-time semantic segmentation approach. In the future, we focus on the application of attention mechanisms to real-time semantic segmentation, including the use of more accurate and efficient backbone networks, as well as proposing faster attention modules.

Acknowledgments

This work was supported in part by the State Key Program of National Natural Science Foundation of China (Grant No.U1908214), in part by the Program for the Liaoning Distinguished Professor, Program for Dalian High-level Talent Innovation Support (Grant No.2017RD11), and in part by the Science and Technology Innovation Fund of Dalian (Grant No. 2020JJ25CY001).

References

Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, jan 2009. doi: 10.1016/j.patrec.2008.04.005. URL https://doi.org/10.1016%2Fj.patrec.2008.04.005.

X. Chen, X. Lou, L. Bai, and J. Han. Residual pyramid learning for single-shot semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2019.

F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2017.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3141–3149, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 603–612, 2019.

Gen Li and Joongkyu Kim. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. In *British Machine Vision Conference*, 2019.

H. Li, P. Xiong, H. Fan, and J. Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9514–9523, 2019a.

X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu. Expectation-maximization attention networks for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9166–9175, 2019b.

Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, and Jing-Jhih Lin. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. In *Proceedings of the ACM Multimedia Asia*, MMAsia '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368414. doi: 10.1145/3338533.3366558. URL https://doi.org/10.1145/3338533.3366558.

Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 561–580, Cham, 2018. Springer International Publishing.

M. Oršic, I. Krešo, P. Bevandic, and S. Šegvic. In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12599–12608, 2019.

Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *ArXiv*, abs/1606.02147, 2016.

E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018.

E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017.

A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

Tianyi Wu, Sheng Tang, Rui Zhang, and Yongdong Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *ArXiv*, abs/1811.08201, 2018.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 334–349, Cham, 2018. Springer International Publishing.

Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 418–434, Cham, 2018a. Springer International Publishing.

Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018b.

Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai. Asymmetric non-local neural networks for semantic segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 593–602, 2019.