CCNet: Criss-Cross Attention for Semantic Segmentation

Zilong Huang[1]*, Xinggang Wang[1], Lichao Huang[2], Chang Huang[2], Yunchao Wei[3], Wenyu Liu[1]
[1]School of EIC, Huazhong University of Science and Technology
[2]Horizon Robotics
[3]Beckman Institute, University of Illinois at Urbana-Champaign

Abstract

Long-range dependencies can capture useful contextual information to benefit visual understanding problems. In this work, we propose a Criss-Cross Network (CCNet) for obtaining such important information through a more effective and efficient way. Concretely, for each pixel, our CCNet can harvest the contextual information of its surrounding pixels on the criss-cross path through a novel criss-cross attention module. By taking a further recurrent operation, each pixel can finally capture the long-range dependencies from all pixels. Overall, our CCNet is with the following merits: 1) GPU memory friendly. Compared with the non-local block, the recurrent criss-cross attention module requires 11× less GPU memory usage. 2) High computational efficiency. The recurrent criss-cross attention significantly reduces FLOPs by about 85% of the non-local block in computing long-range dependencies. 3) The state-of-the-art performance. We conduct extensive experiments on popular semantic segmentation benchmarks including Cityscapes, ADE20K, and instance segmentation benchmark COCO. In particular, our CCNet achieves the mIoU score of 81.4 and 45.22 on Cityscapes test set and ADE20K validation set, respectively, which are the new state-of-the-art results. We make the code publicly available at https://github.com/speedinghzl/CCNet.

1. Introduction

Semantic segmentation is a fundamental topic in computer vision, whose goal is to assign semantic class labels to every pixel in the image. It has been actively studied in many recent papers and is also critical for various challenging applications such as autonomous driving, virtual reality, and image editing.

Recently, state-of-the-art semantic segmentation frameworks based on the fully convolutional network (FCN) [26] have made remarkable progress. Due to the fixed geometric structures, they are inherently limited to local receptive fields and short-range contextual information. These limitations impose a great adverse effect on FCN-based methods due to insufficient contextual information.

To capture long-range dependencies, Chen *et al*. [6] proposed atrous spatial pyramid pooling module with multi-scale dilation convolutions for contextual information ag-

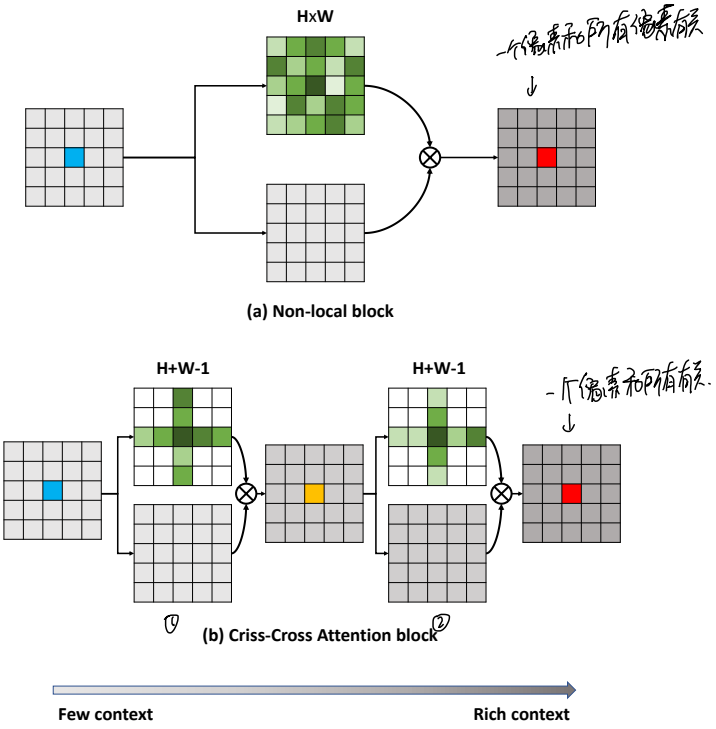

Figure 1. Diagrams of two attention-based context aggregation methods. (a) For each position (*e.g*. blue), Non-local module [32] generates a dense attention map which has $H \times W$ weights (in green). (b) For each position (*e.g*. blue), criss-cross attention module generates a sparse attention map which only has $H + W - 1$ weights. After recurrent operation, each position (*e.g*. red) in the final output feature maps can capture long-range dependencies from all pixels. For clear display, residual connections are ignored.

*The work was mainly done during an internship at Horizon Robotics.

gregation. Zhao *et al*. [42] further introduced PSPNet with pyramid pooling module to capture contextual information. However, the dilated convolution based methods [7, 6, 13] collect information from a few surrounding pixels and can not generate dense contextual information actually. Meanwhile, the pooling based methods [42, 40] aggregate contextual information in a non-adaptive manner and the homogeneous contextual information is adopted by all image pixels, which does not satisfy the requirement the different pixel needs the different contextual dependencies.

To generate dense and pixel-wise contextual information, PSANet [43] learns to aggregate contextual information for each position via a predicted attention map. Non-local Networks [32] utilizes a self-attention mechanism [10, 29], which enable a single feature from any position to perceive features of all the other positions, leading to generate more power pixel-wise representation. Here, each position in the feature map is connected with all other ones through self-adaptively predicted attention maps, thus harvesting various range contextual information, see in Fig. 1 (a). However, these attention-based methods need to generate huge attention maps to measure the relationships for each pixel-pair, whose complexity in time and space are both $\mathcal{O}((H\times W)\times(H\times W))$, where $H\times W$ donates the spatial dimension of input feature map. Since the input feature map is always with high resolution in semantic segmentation task, self-attention based methods have high computation complexity and occupy a huge number of GPU memory. We argue that: Is there an alternative solution to achieve such a target in a more efficient way?

We found that the current non-local operation adopted by [32] can be alternatively replaced by two consecutive criss-cross operations, in which each one only has sparse connections $(H+W-1)$ for each position in the feature maps. This motivates us to propose the criss-cross attention module to aggregate long-range pixel-wise contextual information in horizontal and vertical direction. By serially stacking two criss-cross attention modules, it can collect contextual information from all pixels. The decomposition greatly reduce the complexity in time and space from $\mathcal{O}((H\times W)\times(H\times W))$ to $\mathcal{O}((H\times W)\times(H+W-1))$.

Concretely, our criss-cross attention module is able to harvest various information nearby and far away on the criss-cross path. As shown in Fig. 1, both non-local module, and criss-cross attention module feed the input feature maps with spatial size $H\times W$ to generate attention maps (upper branch) and adapted feature maps (lower branch), respectively. Then, the weighted sum is adopted as aggregation way. In criss-cross attention module, each position (*e.g*., blue color) in the feature map is connected with other ones which are in the same row and the same column through predicted sparsely attention map. The predicted attention map only has $H+W-1$ weights rather than $H\times W$ in non-local module. Furthermore, we propose the recurrent criss-cross attention module to capture the long-range dependencies from all pixels. The local features are passed into criss-cross attention module only once, which collects the contextual information in horizontal and vertical directions. The output feature map of a criss-cross attention module is fed into the next criss-cross attention module; each position (*e.g*. red color) in the second feature map collects information from all others to augment the pixel-wise representations. All the criss-cross attention modules share parameters for reducing extra parameters. Our criss-cross attention module can be plugged into any fully convolutional neural network, named CCNet, for leaning to segment in an end-to-end manner.

We have carried out extensive experiments on large-scale datasets. Our proposed CCNet achieves top performance on two most competitive semantic segmentation datasets, *i.e*., Cityscapes [11], and ADE20K [45]. Besides semantic segmentation, the proposed criss-cross attention even improves the state-of-the-art instance segmentation method, *i.e*., Mask R-CNN with ResNet-101 [16]. The results show that criss-cross attention is generally beneficial to the dense prediction tasks. In summary, our main contributions are two-fold:

- We propose a novel criss-cross attention module in this work, which can be leveraged to capture contextual information from long-range dependencies in a more efficient and effective way.
- We propose a CCNet by taking advantages of two recurrent criss-cross attention modules, achieving leading performance on segmentation-based benchmarks, including Cityscapes, ADE20K and MSCOCO.

2. Related work

Semantic segmentation The last years have seen a renewal of interest on semantic segmentation. FCN [26] is the first approach to adopt fully convolution network for semantic segmentation. Later, FCN-based methods have made great progress in image semantic segmentation. Chen *et al*. [5] and Yu *et al*. [38] removed the last two downsample layers to obtain dense prediction and utilized dilated convolutions to enlarge the receptive field. Unet [28], Deeplabv3+ [9], RefineNet [21] and DFN [37] adopted encoder-decoder structures that fuse the information in low-level and high-level layers to predict segmentation mask. SAC [41] and Deformable Convolutional Networks [12] improved the standard convolution operator to handle the deformation and various scale of objects. CRF-RNN [38] and DPN [25] used Graph model *i.e*. CRF,MRF, for semantic segmentation. AAF [19] used adversarial learning

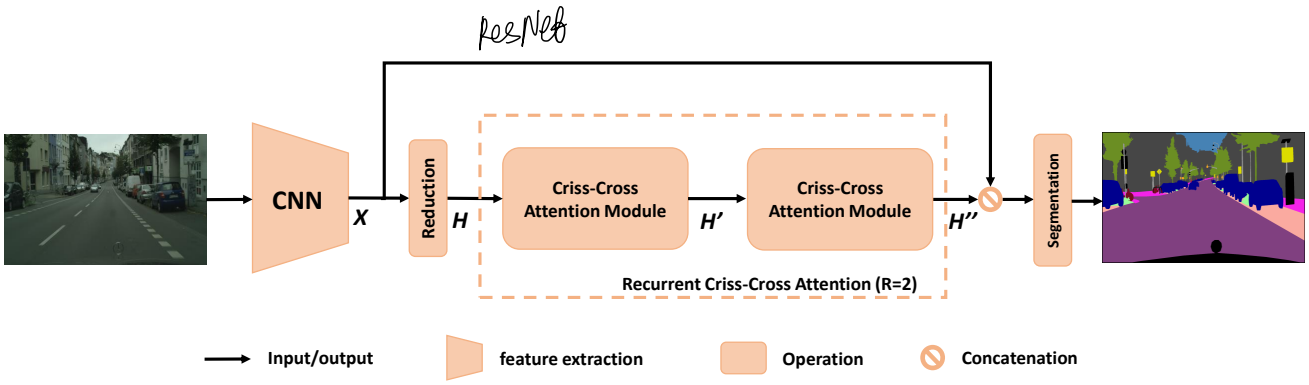

Figure 2. Overview of the proposed CCNet for semantic segmentation. The proposed recurrent criss-cross attention takes as input feature maps $\mathbf{H}$ and output feature maps $\mathbf{H}''$ which obtain rich and dense contextual information from all pixels. Recurrent criss-cross attention module can be unrolled into $R = 2$ loops, in which all Criss-Cross Attention module share parameters.

to capture and match the semantic relations between neighboring pixels in the label space. BiSeNet [36] was designed for real-time semantic segmentation.

In addition, some works aggregate the contextual information to augment the feature representation. Deeplabv2 [6] proposed ASPP module to use different dilation convolutions to capture contextual information. DenseASPP [35] brought dense connections into ASPP to generate features with various scale. DPC [4] utilized architecture search techniques to build multi-scale architectures for semantic segmentation. PSPNet [42] utilized pyramid pooling to aggregate contextual information. GCN [27] utilized global convolution module and ParseNet [24] utilized global pooling to harvest context information for global representations. Recently, Zhao *et al*. [43] proposed the pointwise spatial attention network which uses predicted attention map to guide contextual information collection. Liu *et al*. [23] and Visin *et al*. [30] utilized RNNs to capture long-range contextual dependency information. conditional random field (CRF) [2, 3, 5, 44], Markov random field (MRF) [25] and recurrent neural network (RNN) [23] are also utilized to capture long-range dependencies.

Attention model Attention model is widely used for various tasks. Squeeze-and-Excitation Networks [17] enhanced the representational power of the network by modeling channel-wise relationships in an attention mechanism. Chen *et al*. [8] made use of several attention masks to fuse feature maps or predictions from different branches. Vaswani *et al*. [29] applied a self-attention model on machine translation. Wang *et al*. [32] proposed the non-local module to generate the huge attention map by calculating the correlation matrix between each spatial point in the feature map, then the attention guided dense contextual information aggregation. OCNet [39] and DANet [14] utilized self-attention mechanism to harvest the contextual information. PSA [43] learned an attention map to aggregate contextual information for each individual point adaptively and specifically. Our CCNet is different from the aforementioned studies to generate huge attention map to record the relationship for each pixel-pair in feature map. The contextual information is aggregated by criss-cross attention module on the criss-cross path. Beside, CCNet can also obtain dense contextual information in a recurrent fashion which is more effective and efficient.

3. Approach

In this section, we give the details of the proposed Criss-Cross Network (CCNet) for semantic segmentation. At first, we will first present a general framework of our network. Then, we will introduce criss-cross attention module which captures long-range contextual information in horizontal and vertical direction. At last, to capture the dense and global contextual information, we propose the recurrent criss-cross attention module.

3.1. Overall

The network architecture is given in Fig. 2. An input image is passed through a deep convolutional neural networks (DCNN), which is designed in a fully convolutional fashion [6], then, produces a feature map $\mathbf{X}$. We denote the spatial size of $\mathbf{X}$ as $H \times W$. In order to retain more details and efficiently produce dense feature maps, we remove the last two down-sampling operations and employ dilation convolutions in the subsequent convolutional layers, thus enlarging the width/height of the output feature maps $\mathbf{X}$ to 1/8 of the input image.

After obtaining feature maps $\mathbf{X}$, we first apply a convolution layer to obtain the feature maps $\mathbf{H}$ of dimension reduction, then, the feature maps $\mathbf{H}$ would be fed into the criss-cross attention (CCA) module and generate new feature maps $\mathbf{H}'$ which aggregate long-range contextual infor-

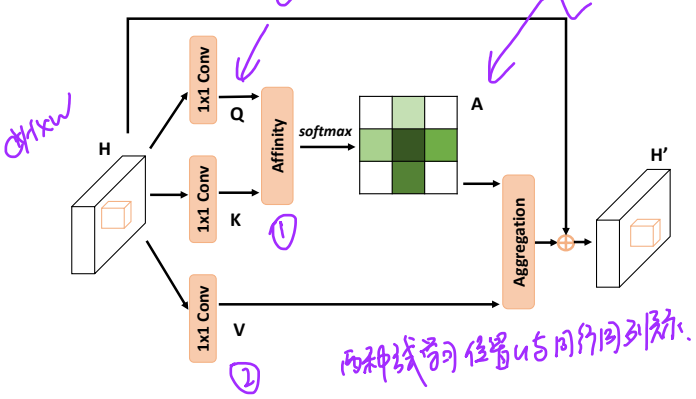

Figure 3. The details of criss-cross attention module.

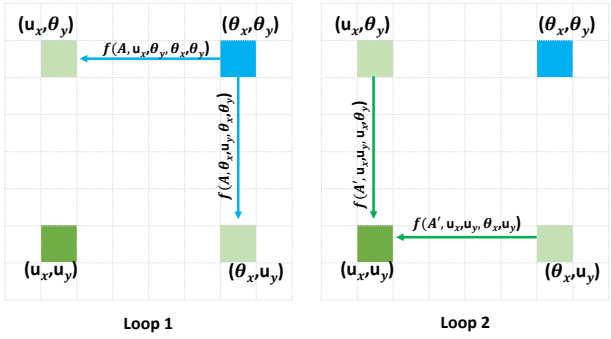

Figure 4. An example of information propagation when the loop number is 2.

mation together for each pixel in a criss-cross way. The feature maps $\mathbf{H}'$ only aggregate the contextual information in horizontal and vertical directions which are not powerful enough for semantic segmentation. To obtain richer and denser context information, we feed the feature maps $\mathbf{H}'$ into the criss-cross attention module again and output feature maps $\mathbf{H}''$. Thus, each position in feature maps $\mathbf{H}''$ actually gather the information from all pixels. Two criss-cross attention modules before and after share the same parameters to avoid adding too many extra parameters. We name this recurrent structure as recurrent criss-cross attention (RCCA) module.

Then we concatenate the dense contextual feature $\mathbf{H}''$ with the local representation feature $\mathbf{X}$. It is followed by one or several convolutional layers with batch normalization and activation for feature fusion. Finally, the fused features are fed into the segmentation layer to generate the final segmentation map.

3.2. Criss-Cross Attention

In order to model long-range contextual dependencies over local feature representations using lightweight computation and memory, we introduce a criss-cross attention module. The criss-cross attention module collects contextual information in horizontal and vertical directions to enhance pixel-wise representative capability.

As shown in Fig 3, given a local feature $\mathbf{H} \in \mathbb{R}^{C\times W\times H}$, the criss-cross attention module firstly applies two convolution layers with 1×1 filters on $\mathbf{H}$ to generate two feature maps $\mathbf{Q}$ and $\mathbf{K}$, respectively, where $\{\mathbf{Q}, \mathbf{K}\} \in \mathbb{R}^{C'\times W\times H}$. C' is the channel number of feature maps, which is less than C for dimension reduction.

After obtaining feature maps $\mathbf{Q}$ and $\mathbf{K}$, we further generate attention maps $\mathbf{A} \in \mathbb{R}^{(H+W-1)\times W\times H}$ via **Affinity** operation. At each position u in spatial dimension of feature maps $\mathbf{Q}$, we can get a vector $\mathbf{Q_u} \in \mathbb{R}^{C'}$. Meanwhile, we can obtain the set $\mathbf{\Omega_u}$ by extracting feature vectors from $\mathbf{K}$ which are in the same row or column with position u. Thus, $\mathbf{\Omega_u} \in \mathbb{R}^{(H+W-1)\times C'}$. $\mathbf{\Omega_{i,u}} \in \mathbb{R}^{C'}$ is ith element of $\mathbf{\Omega_u}$. The **Affinity** operation is defined as follows:

$$d_{i,u} = \mathbf{Q_u}\mathbf{\Omega_{i,u}}^{\mathsf{T}} \quad (1)$$

in which $d_{i,u} \in \mathbf{D}$ denotes the degree of correlation between feature $\mathbf{Q_u}$ and $\mathbf{\Omega_{i,u}}$, $i = [1, ..., |\mathbf{\Omega_u}|]$, $\mathbf{D} \in \mathbb{R}^{(H+W-1)\times W\times H}$. Then, we apply a softmax layer on $\mathbf{D}$ along the channel dimension to calculate the attention map $\mathbf{A}$.

Then another convolutional layer with 1×1 filters is applied on $\mathbf{H}$ to generate $\mathbf{V} \in \mathbb{R}^{C\times W\times H}$ for feature adaption. At each position u in spatial dimension of feature maps $\mathbf{V}$, we can obtain a vector $\mathbf{V_u} \in \mathbb{R}^{C}$ and a set $\mathbf{\Phi_u} \in \mathbb{R}^{(H+W-1)\times C}$. The set $\mathbf{\Phi_u}$ is collection of feature vectors in $\mathbf{V}$ which are in the same row or column with position u. The long-range contextual information is collected by the **Aggregation** operation:

$$\mathbf{H'_u} = \sum_{i\in|\mathbf{\Phi_u}|} \mathbf{A_{i,u}}\mathbf{\Phi_{i,u}} + \mathbf{H_u} \quad (2)$$

in which $\mathbf{H'_u}$ denotes a feature vector in output feature maps $\mathbf{H}' \in \mathbb{R}^{C\times W\times H}$ at position u. $\mathbf{A_{i,u}}$ is a scalar value at channel i and position u in $\mathbf{A}$. The contextual information is added to local feature $\mathbf{H}$ to enhance the local features and augment the pixel-wise representation. Therefore, it has a wide contextual view and selectively aggregates contexts according to the spatial attention map. This feature representations achieve mutual gains and are more robust for semantic segmentation.

The proposed criss-cross attention module is a self-contained module which can be dropped into a CNN architecture at any point, and in any number, obtaining rich contextual information. This module is very computationally cheap and adds a few parameters, causing very little GPU memory usage.

3.3. Recurrent Criss-Cross Attention

Despite a criss-cross attention module can capture long-range contextual information in horizontal and vertical direction, the connections between the pixel and around pixels are still sparse. It is helpful to obtain dense contextual information for semantic segmentation. To achieve this, we introduce the recurrent criss-cross attention based on the criss-cross attention module described above. The recurrent criss-cross attention module can be unrolled into R loops. In the first loop, the criss-cross attention module takes as input feature maps $\mathbf{H}$ extracted from a CNN model and output feature maps $\mathbf{H}'$, where $\mathbf{H}$ and $\mathbf{H}'$ have the same shape. In the second loop, the criss-cross attention module takes as input feature maps $\mathbf{H}'$ and output feature maps $\mathbf{H}''$. As shown in Fig. 2, recurrent criss-cross attention module has two loops (R=2) which is enough to harvest long-range dependencies from all pixels to generate new feature maps with dense and rich contextual information.

The $\mathbf{A}$ and $\mathbf{A}'$ are donated as the attention maps in loop 1 and loop 2, respectively. Since we are interested only in contextual information spreads in spatial dimension rather than in channel dimension, the convolutional layer with 1×1 filters can be view as identical connection. In addition, the mapping function from position x', y' to weight $A_{i,x,y}$ is defined as $A_{i,x,y} = f(A, x, y, x', y')$. For any position u at feature map $\mathbf{H}''$ and any position θ at feature map $\mathbf{H}$, there is a connection if $R = 2$. One case is that u and θ are in the same row or column:

$$\mathbf{H}''_{\mathbf{u}} \leftarrow [f(A, u, \theta) + 1] \cdot f(A', u, \theta) \cdot \mathbf{H}_{\theta} \tag{3}$$

in which $\leftarrow$ donates add-to operation. Another case is that u and θ are not in the same row and column. Fig 4 shows the propagation path of context information in spatial dimension:

$$\begin{aligned}\mathbf{H}''_{\mathbf{u}} \leftarrow [&f(A, u_x, \theta_y, \theta_x, \theta_y) \cdot f(A', u_x, u_y, u_x, \theta_y) + \\ &f(A, \theta_x, u_y, \theta_x, \theta_y) \cdot f(A', u_x, u_y, \theta_x, u_y)] \cdot \mathbf{H}_{\theta}\end{aligned} \tag{4}$$

In general, our recurrent criss-cross attention module makes up for the deficiency of criss-cross attention module that cannot obtain the dense contextual information from all pixels. Compared with criss-cross attention module, the recurrent criss-cross attention module ($R = 2$) does not bring extra parameters and can achieve better performance with the cost of minor computation increment. The recurrent criss-cross attention module is also a self-contained module that can be plugged into any CNN architecture at any stage and be optimized in an end-to-end manner.

4. Experiments

To evaluate the proposed method, we carry out comprehensive experiments on Cityscapes dataset, ADE20K dataset, and COCO dataset. Experimental results demonstrate that CCNet achieves state-of-the-art performance on Cityscapes and ADE20K. Meanwhile, CCNet can bring constant gain on COCO for instance segmentation. In the following subsections, we first introduce the datasets and implementation details, then we perform a series of ablation experiments on Cityscapes dataset. Finally, we report our results on ADE20K and COCO dataset.

4.1. Datasets and Evaluation Metrics

We adopt Mean IoU (mean of class-wise intersection over union) for Cityscapes and ADE20K and standard COCO metrics Average Precision (AP) for COCO.

- **Cityscapes** is tasked for urban segmentation, which contains 5,000 high quality pixel-level finely annotated images and 20,000 coarsely annotated images captured from 50 different cities. Each image is with 1024×2048 resolution, which has 19 classes for semantic segmentation evaluation. Only the 5,000 finely annotated images are used in our experiments and are divided into 2,975/500/1,525 images for training, validation, and testing.

- **ADE20K** is a recent scene parsing benchmark containing dense labels of 150 stuff/object categories. The dataset includes 20K/2K/3K images for training, validation and test.

- **COCO** is a very challenging dataset that contains 115K images over 80 categories for training, 5K for validation and 20k for testing.

4.2. Implementation Details

Network Structure We implement our method based on open source pytorch segmentation toolbox [18]. For semantic segmentation, we choose the ImageNet pre-trained ResNet-101 [16] as our backbone and remove the last two down-sampling operations and employ dilated convolutions in the subsequent convolutional layers following the previous works [5], the output stride becomes 8. Meanwhile, we replace the standard Batchnorm with InPlace-ABN [1] to the mean and standard-deviation of BatchNorm across multiple GPUs. For instance segmentation, we choose Mask-RCNN [15] as our baseline.

Training settings The SGD with mini-batch is used for training. For semantic segmentation, the initial learning rate is 1e-2 for Cityscapes and ADE20K. Following prior works [6, 40], we employ a poly learning rate policy where the initial learning rate is multiplied by $1 - (\frac{iter}{max\_iter})^{power}$ with power = 0.9. We use the momentum of 0.9 and a weight decay of 0.0001. For Cityscapes, the training images are augmented by randomly scaling (from 0.75 to 2.0),

Table 1. Comparison with state-of-the-arts on Cityscapes validation set.

Method	Backbone	multi-scale	mIOU(%)
DeepLabv3 [7]	ResNet-101	Yes	79.3
DeepLabv3+ [9]	Xception-65	No	79.1
DPC [4] †	Xception-71	No	80.8
CCNet	ResNet-101	Yes	**81.3**

† use extra COCO dataset for training.

then randomly cropping out the high-resolution patches (769×769) from the resulting images. Since the images from ADE20K are with various sizes, we adopt an augmentation strategy of resizing the short side of input image to the length randomly chosen from the set {300, 375, 450, 525, 600}. In addition, we also apply random flipping horizontally for data augmentation. We employ 4 $\times$ TITAN XP GPUs for training and batch size is 8. For instance segmentation, we take the same training settings as that of Mask-RCNN [15].

4.3. Experiments on Cityscapes

4.3.1 Comparisons with state-of-the-arts

Results of other state-of-the-art semantic segmentation solutions on cityscapes validation set are summarized in Tab. 1. We provide these results for reference and emphasize that they should not be directly compared with our method. Among the approaches, Deeplabv3 [7] and CCNet uses the same backbone and multi-scale testing strategy. Deeplabv3+ [9] and DPC [4] use more stronger backbone. In particular, DPC [4] make use of COCO dataset for training rather Cityscapes training set. The results show that the proposed CCNet with multi-scale testing achieves the new state-of-the-art performance.

In addition, we also train the best learned CCNet with ResNet-101 [16] as the backbone using both training and validation sets and make the evaluation on the test set by submitting our test results to the official evaluation server. Most of methods [6, 21, 41, 27, 31, 42, 36, 19, 43, 37] adopt the same backbone as ours and the others [33, 35] utilize stronger backbones. From Tab. 2, it can be observed that our CCNet substantially outperforms all the previous techniques. Among the approaches, PSANet [43] is most related to our method which generates sub attention map for each pixel. One of the differences is that the sub attention map has $2 \times H \times W$ weights in PSANet and $H + W - 1$ weights in CCNet. Our method can achieve better performance with low computation cost and low memory usage.

Table 2. Cityscapes test set performance across leading competitive models.

Method	Backbone	mIOU(%)
DeepLab-v2 [6]	ResNet-101	70.4
RefineNet [21] ‡	ResNet-101	73.6
SAC [41] ‡	ResNet-101	78.1
GCN [27] ‡	ResNet-101	76.9
DUC [31] ‡	ResNet-101	77.6
ResNet-38 [33]	WiderResnet-38	78.4
PSPNet [42]	ResNet-101	78.4
BiSeNet [36] ‡	ResNet-101	78.9
AAF [19] ‡	ResNet-101	79.1
PSANet [43] ‡	ResNet-101	80.1
DFN [37] ‡	ResNet-101	79.3
DenseASPP [35] ‡	DenseNet-161	80.6
CCNet ‡	ResNet-101	**81.4**

‡ train with both the train-fine and val-fine datasets.

4.3.2 Ablation studies

To further prove the effectiveness of the CCNet, we conduct extensive ablation experiments on the validation set of Cityscapes with different settings for CCNet.

The effect of attention module Tab. 3 demonstrates the performance on Cityscapes validation set by adopting different number of recurrent criss-cross attention module (RCCA). All experiments are conducted using Resnet-101 as the backbone. Beside, the input size of image is 769×769, resulting in the size of input feature map H of RCCA is 97×97. Our baseline network is ResNet-based FCN with dilated convolution module incorporated at stage 4 and 5, *i.e.*, dilations are set to 2 and 4 for these two stages respectively. The increment of FLOPs and Memory usage are estimated when $R = 1, 2, 3$, respectively. We can observe that adding a criss-cross attention into the baseline, donated as $R = 1$, improves the performance by 2.9% compared with the baseline, which can effectively demonstrate the significance of criss-cross attention module. Furthermore, increasing loops from 1 to 2 can improve the performance by 1.8%, demonstrating the effectiveness of dense contextual information. Finally, increasing loops from 2 to 3 slightly improves the performance by 0.4%. Meanwhile, with the increasing of loops, the usage of FLOPs and GPU memory will still be increased. These results prove that the proposed criss-cross attention module can significantly improve the performance by capturing long-range contextual information in horizontal and vertical direction. In addition, the proposed criss-cross attention is effective in capturing the dense and global contextual information, which can finally

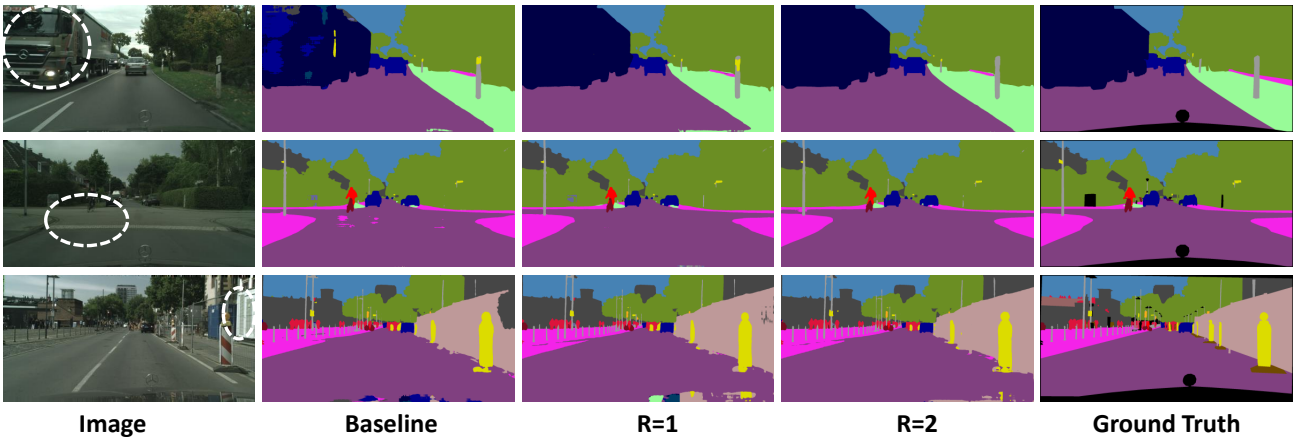

Figure 5. Visualization results of RCCA with different loops on Cityscapes validation set.

benefit the performance of semantic segmentation. To balance the performance and resource usage, we choose $R = 2$ as default settings in all the following experiments.

We provide the qualitative comparisons in Fig. 5 to further validate the effectiveness of the criss-cross module. We leverage the *white circles* to indicate those challenging regions that are easily to be misclassified. We can observe that these challenging regions are progressively corrected with the increasing of loops, which can well prove the effectiveness of dense contextual information aggregation for semantic segmentation.

Comparison of context aggregation approaches We compare the performance of several different context aggregation approaches on the Cityscapes validation set with Resnet-50 and Resnet-101 as backbones. It should be noted that we do not provide the result of "Resnet-101 + NL", because we can not run the experiment that integrates non-local block into Resnet-101 backbone due to the limitation of 12G GPU memory.

Specifically, the baselines of context aggregation mainly include: 1) Zhao *et al*. [42] proposed Pyramid pooling which is the simple and effective way to capture global contextual information, donated as "+PP"; 2) Chen *et al*. [7] used different dilation convolutions to harvest pixel-wise contextual information at the different range, donated as "+ASPP"; 3) Wang *et al*. [32] introduced non-local network whose attention mask for each position is generated by calculating the feature correlation between each pixel-pair to guide context aggregation, donated as "+NL".

In Tab. 4, Both "+NL" and "+RCCA" achieve better performance compared with other the context aggregation approaches, which demonstrates the importance of capturing dense long-range contextual information. More interestingly, our method achieves better performance than "+NL" which can also form dense long-range contextual information. One cause may be that the attention map plays a key role for contextual information aggregation. "+NL" generates an attention map from the feature which has limit receptive field and short range contextual information, but our "+RCCA" takes two steps to form dense contextual information, leading to that the latter step can learn a better attention map benefiting from the feature map produced by the first step in which some long rage contextual information has already been embedded.

We further explore the amount of computation and memory footprint of RCCA. As shown in Table 5, compared with "+NL" method, the proposed "+RCCA" requires $11\times$ less GPU memory usage and significantly reduce FLOPs by about 85% of non-local block in computing long-range dependencies, which shows that the CCNet is an efficient way to capture long-range contextual information in the least amount of computation and memory footprint.

Visualization of Attention Map To get a deeper understanding of our RCCA, we visualize the learned attention masks as shown in Fig. 6. For each input image, we select one point (green cross) and show its corresponding attention maps when $R = 1$ and $R = 2$ in columns 2 and 3 respectively. From Fig. 6, only contextual information from the criss-cross path of the target point is capture when $R = 1$. By adopting one more criss-cross module, *i.e*., $R = 2$ the RCCA can finally aggregate denser and richer contextual information compared with that of $R = 1$. Besides, we observe that the attention module could capture semantic similarity and long-range dependencies.

4.4. Experiments on ADE20K

In this subsection, we conduct experiments on the AED20K dataset, which is very challenging segmentation dataset for both indoor and outdoor scene parsing, to validate the effectiveness of our method. As shown in Tab. 6, the CCNet achieves the state-of-the-art performance of 45.22%, outperforms the previous state-of-the-art meth-

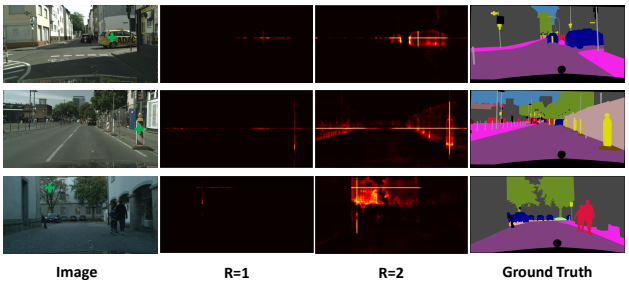

Figure 6. Visualization results of attention module on Cityscapes validation set. The left column is the images from the validation set of Cityscapes, the 2 and 3 columns are pixel-wise attention maps when $R = 1$ and $R = 2$ in RCCA. The last column is ground truth.

Table 3. Performance on Cityscapes validation set for different loops in RCCA. FLOPs and Memory usage are estimated for an input of $1 \times 3 \times 769 \times 769$.

Loops	GFLOPs(▲)	Memory(M▲)	mIOU(%)
baseline	0	0	75.1
R=1	8.3	53	78.0
R=2	16.5	127	79.8
R=3	24.7	208	80.2

Table 4. Comparison of context aggregation approaches on Cityscapes validation set.

Method	mIOU(%)
ResNet50-Baseline	73.3
ResNet50+PSP	76.4
ResNet50+ASPP	77.1
ResNet50+NL	77.3
ResNet50+RCCA(R=2)	78.5
ResNet101-Baseline	75.1
ResNet101+PSP	78.5
ResNet101+ASPP	78.9
ResNet101+RCCA(R=2)	79.8

Table 5. Comparison of Non-local module and RCCA. FLOPs and Memory usage are estimated for an input of $1 \times 3 \times 769 \times 769$.

Method	GFLOPs(▲)	Memory(M▲)	mIOU(%)
baseline	0	0	73.3
+NL	108	1411	77.3
+RCCA(R=2)	16.5	127	78.5

ods by more than 0.6%. Among the approaches, most of methods [41, 42, 43, 20, 34, 40] adopt the ResNet-101 as backbone and RefineNet [21] adopts a more powerful network, *i.e.*, ResNet-152, as the backbone. EncNet [40] achieves previous best performance among the methods and utilizes global pooling with image-level supervision to collect image-level context information. In contrast, our CCNet adopts an alternative way to integrate contextual information by capture pixel-wise long-range dependencies and achieve better performance.

Table 6. State-of-the-art Comparison experiments on ADE20K validation set.

Method	Backbone	mIOU(%)
RefineNet [21]	ResNet-152	40.70
SAC [41]	ResNet-101	44.30
PSPNet [42]	ResNet-101	43.29
PSANet [43]	ResNet-101	43.77
DSSPN [20]	ResNet-101	43.68
UperNet [34]	ResNet-101	42.66
EncNet [40]	ResNet-101	44.65
CCNet	ResNet-101	**45.22**

Table 7. Results of object detection and instance segmentation on COCO.

Method		AP^{box}	AP^{mask}
R50	baseline	38.2	34.8
	+NL	39.0	35.5
	+RCCA	**39.3**	**36.1**
R101	baseline	40.1	36.2
	+NL	40.8	37.1
	+RCCA	**41.0**	**37.3**

4.5. Experiments on COCO

To further demonstrate the generality of our CCNet, We conduct the instance segmentaion task on COCO [22] using the competitive Mask R-CNN model [15] as the baseline. Following [32], we modify the Mask R-CNN backbone by adding the RCCA module right before the last convolutional residual block of res4. We evaluate a standard baseline of ResNet-50/101. All models are fine-tuned from ImageNet pre-training. We use open source implementation[1] with end-to-end joint training whose performance is almost the same as the baseline reported in [32]. We report the comparisons in terms of box AP and mask AP in Tab. 7 on COCO. The results demonstrate that our method substantially outperforms the baseline in all metrics. Meanwhile, the network with "+RCCA" also achieve the better performance than the network with one non-local block "+NL".

[1] https://github.com/facebookresearch/maskrcnn-benchmark

5. Conclusion and future work

In this paper, we have presented a Criss-Cross Network (CCNet) for semantic segmentation, which adaptively captures long-range contextual information on the criss-cross path. To obtain dense contextual information, we introduce recurrent criss-cross attention module which aggregates contextual information from all pixels. The ablation experiments demonstrate that recurrent criss-cross attention captures dense long-range contextual information in less computation cost and less memory cost. Our CCNet achieves outstanding performance consistently on two semantic segmentation datasets, *i.e*. Cityscapes, ADE20K and instance segmentation dataset, *i.e*. COCO.

References

[1] S. R. Bulò, L. Porzi, and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. *CoRR, abs/1712.02616, December*, 5, 2017.

[2] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *European Conference on Computer Vision*, pages 402–418. Springer, 2016.

[3] S. Chandra, N. Usunier, and I. Kokkinos. Dense and low-rank gaussian crfs using deep embeddings. In *International Conference on Computer Vision (ICCV)*, volume 2, 2017.

[4] L.-C. Chen, M. D. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens. Searching for efficient multi-scale architectures for dense image prediction. *arXiv preprint arXiv:1809.04184*, 2018.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[9] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[10] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[11] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[12] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR, abs/1703.06211*, 1(2):3, 2017.

[13] H. Ding, X. Jiang, B. Shuai, A. Qun Liu, and G. Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

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

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.

[18] Z. Huang, Y. Wei, X. Wang, and W. Liu. A pytorch semantic segmentation toolbox. https://github.com/speedinghzl/pytorch-segmentation-toolbox, 2018.

[19] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu. Adaptive affinity field for semantic segmentation. *arXiv preprint arXiv:1803.10335*, 2018.

[20] X. Liang, H. Zhou, and E. Xing. Dynamic-structured semantic propagation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 752–761, 2018.

[21] G. Lin, A. Milan, C. Shen, and I. D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Cvpr*, volume 1, page 5, 2017.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems*, pages 1520–1530, 2017.

[24] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[25] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015.

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

[27] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel mattersimprove semantic segmentation by global convolutional network. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1743–1751. IEEE, 2017.

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

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[30] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.

[31] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. IEEE, 2018.

[32] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *arXiv preprint arXiv:1711.07971*, 10, 2017.

[33] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[34] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. *arXiv preprint arXiv:1807.10221*, 2018.

[35] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3684–3692, 2018.

[36] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. *arXiv preprint arXiv:1808.00897*, 2018.

[37] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. *arXiv preprint arXiv:1804.09337*, 2018.

[38] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[39] Y. Yuan and J. Wang. Ocnet: Object context network for scene parsing. *arXiv preprint arXiv:1809.00916*, 2018.

[40] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *Proc. 26th Int. Conf. Comput. Vis.*, pages 2031–2039, 2017.

[42] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[43] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision*, pages 270–286. Springer, 2018.

[44] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015.

[45] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 4. IEEE, 2017.