CCNet: Criss-Cross Attention for Semantic Segmentation

Zilong Huang, Xinggang Wang, *Member, IEEE*, Yunchao Wei, Lichao Huang, Humphrey Shi, *Member, IEEE*, Wenyu Liu, *Senior Member, IEEE*, and Thomas S. Huang, *Life Fellow, IEEE*

Abstract—Contextual information is vital in visual understanding problems, such as semantic segmentation and object detection. We propose a Criss-Cross Network (CCNet) for obtaining full-image contextual information in a very effective and efficient way. Concretely, for each pixel, a novel criss-cross attention module harvests the contextual information of all the pixels on its criss-cross path. By taking a further recurrent operation, each pixel can finally capture the full-image dependencies. Besides, a category consistent loss is proposed to enforce the criss-cross attention module to produce more discriminative features. Overall, CCNet is with the following merits: 1) GPU memory friendly. Compared with the non-local block, the proposed recurrent criss-cross attention module requires $11\times$ less GPU memory usage. 2) High computational efficiency. The recurrent criss-cross attention significantly reduces FLOPs by about 85% of the non-local block. 3) The state-of-the-art performance. We conduct extensive experiments on semantic segmentation benchmarks including Cityscapes, ADE20K, human parsing benchmark LIP, instance segmentation benchmark COCO, video segmentation benchmark CamVid. In particular, our CCNet achieves the mIoU scores of 81.9%, 45.76% and 55.47% on the Cityscapes test set, the ADE20K validation set and the LIP validation set respectively, which are the new state-of-the-art results. The source codes are available at https://github.com/speedinghzl/CCNet.

Index Terms—Semantic Segmentation, Graph Attention, Criss-Cross Network, Context Modeling

1 Introduction

Semantic segmentation, which is a fundamental problem in the computer vision community, aims at assigning semantic class labels to each pixel in a given image. It has been extensively and actively studied in many recent works and is also critical for various significant applications such as autonomous driving [1], augmented reality [2],image editing [3], civil engineering [4], remote sensing imagery [5] and agricultural pattern analysis [6], [7]. Specifically, current state-of-the-art semantic segmentation approaches based on the fully convolutional network (FCN) [8] have made remarkable progress. However, due to the fixed geometric structures, the conventional FCN is inherently limited to local receptive fields that only provide short-range contextual information. The limitation of insufficient contextual information imposes a great adverse effect on its segmentation accuracy.

To make up for the above deficiency of FCN, some works have been proposed to introduce useful contextual information to benefit the semantic segmentation task. Specifically, Chen *et al.* [10] proposed atrous spatial pyramid pooling

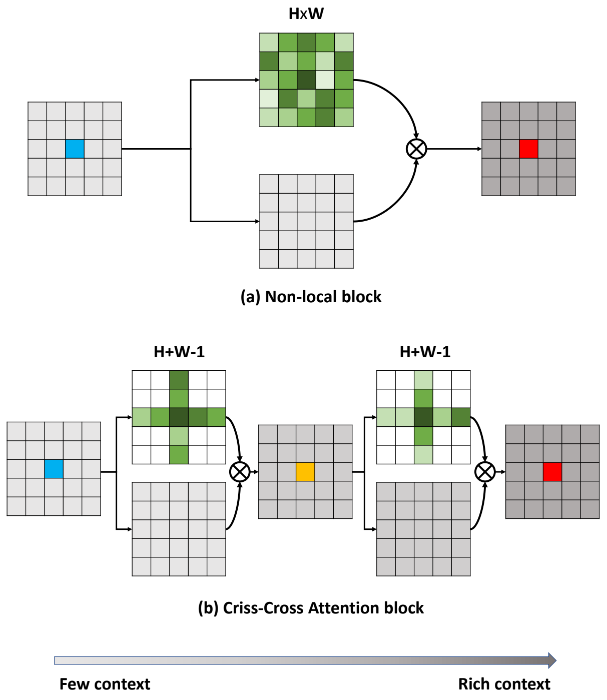

Fig. 1. Diagrams of two attention-based context aggregation methods. (a) For each position (*e.g.*, blue), the Non-local module [9] generates a dense attention map which has N weights (in green). (b) For each position (*e.g.*, blue), the criss-cross attention module generates a sparse attention map which only has about $2\sqrt{N}$ weights. After the recurrent operation, each position (*e.g.*, red) in the final output feature maps can collect information from all pixels. For clear display, residual connections are ignored.

- Z. Huang, X. Wang and W. Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: hzl@hust.edu.cn, xgwang@hust.edu.cn, liuwy@hust.edu.cn).
- Y. Wei is with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia. (e-mail: yunchao.wei@uts.edu.au).
- L. Huang is with Horizon Robotics. (e-mail: lichao.huang@horizon.ai).
- H. Shi is with the University of Oregon and the University of Illinois at Urbana-Champaign. (e-mail: shihonghui3@gmail.com).
- T. S. Huang was with the University of Illinois at Urbana-Champaign. (e-mail: t-huang1@illinois.edu).

Corresponding author: Xinggang Wang. Zilong Huang and Xinggang Wang contributed equally to this work.

module with multi-scale dilation convolutions for contextual information aggregation. Zhao *et al*. [11] further introduced PSPNet with pyramid pooling module to capture contextual information. However, the dilated convolution based methods [10], [12], [13] collect information from a few surrounding pixels and cannot generate dense contextual information actually. Meanwhile, the pooling based methods [11], [14] aggregate contextual information in a non-adaptive manner and the homogeneous context extraction procedure is adopted by all image pixels, which does not satisfy the requirement that different pixels need different contextual dependencies.

To incorporate dense and pixel-wise contextual information, some fully-connected graph neural network (GNN) [15] methods were proposed to augments traditional convolutional features with an estimated full-image context representation. PSANet [16] learns to aggregate contextual information for each position via a predicted attention map. Non-local Networks [9] utilizes a self-attention mechanism [17], [18], which enables a single feature from any position to perceive features of all the other positions, thus harvesting full-image contextual information, see Fig. 1 (a). These non-local operations could be viewed as a densely-connected GNN module based on attention mechanism [18]. This feature augmentation method allows a flexible way to represent non-local relations between features and has led to significant improvements in several vision recognition tasks. However, these GNN-based non-local neural networks need to generate huge attention maps to measure the relationships for each pixel-pair, leading to a very high complexity of $\mathcal{O}(N^2)$ for both time and space, where N is the number of input features. Since the dense prediction tasks, such as semantic segmentation, inherently require high resolution feature maps, the non-local based methods will often with high computation complexity and occupy a huge number of GPU memory. Thus, is there an alternative solution to achieve such a target in a more efficient way?

To address the above mentioned issue, our motivation is to replace the common single densely-connected graph with several consecutive sparsely-connected graphs, which usually require much lower computational resources. Without loss of generality, we use two consecutive criss-cross attention modules, in which each one only has sparse connections (about $\sqrt{N}$) for each position in the feature map. For each pixel/position, the criss-cross attention module aggregates contextual information in its horizontal and vertical directions. By serially stacking two criss-cross attention modules, each position can collect contextual information from all pixels in the given image. The above decomposition strategy will greatly reduce the complexities of both time and space from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\sqrt{N})$.

We compare the differences between the non-local module [9] and our criss-cross attention module in Fig. 1. Concretely, both non-local module and criss-cross attention module feed the input feature map to generate an attention map for each position and transform the input feature map into an adapted feature map. Then, a weighted sum is adopted to collecting contextual information from other positions in the adapted feature map based on the attention maps. Different from the dense connections adopted by the non-local module, each position (*e.g.*, blue) in the feature map is sparsely connected with other ones which are in the same row and the same column in our criss-cross attention module, leading to the predicted attention map only has about $2\sqrt{N}$ weights rather than N in non-local module.

To achieve the goal of capturing the full-image dependencies, we innovatively and simply take a recurrent operation for the criss-cross attention module. In particular, the local features are firstly passed through one criss-cross attention module to collect the contextual information in horizontal and vertical directions. Then, by feeding the feature map produced by the first criss-cross attention module into the second one, the additional contextual information obtained from the criss-cross path finally enables the full-image dependencies for all positions. As demonstrated in Fig. 1 (b), each position (*e.g.*red) in the second feature map can collect information from all others to augment the position-wise representations. We share parameters of the criss-cross modules to keep our model slim. Since the input and output are both convolutional feature maps, criss-cross attention module can be easily plugged into any fully convolutional neural network, named as CCNet, for learning full-image contextual information in an end-to-end manner. Thanks to the good usability of criss-cross attention module, CCNet is straight forward to extend to 3D networks for capturing long-range temporal context information.

In addition, to drive the proposed recurrent criss-cross attention method to learn more discriminative features, we introduce a category consistent loss to augment CCNet. Particularly, the category consistent loss enforces the network to map each pixel in the image to an n-dimensional vector in the feature space, such that feature vectors of pixels that belong to the same category lie close together while feature vectors of pixels that belong to different categories lie far apart.

We have carried out extensive experiments on multiple large-scale datasets. Our proposed CCNet achieves top performance on four most competitive semantic segmentation datasets, *i.e.*, Cityscapes [19], ADE20K [20], LIP [21] and CamVid [22]. In addition, the proposed criss-cross attention even improves the state-of-the-art instance segmentation method, *i.e.*, Mask R-CNN with ResNet-101 [23]. These results well demonstrate that our criss-cross attention module is generally beneficial to the dense prediction tasks. In summary, our main contributions are three-fold:

- We propose a novel criss-cross attention module in this work, which can be leveraged to capture contextual information from full-image dependencies in a more efficient and effective way.
- We propose category consistent loss which can enforce criss-cross attention module to produce more discriminative features.
- We propose CCNet by taking advantages of recurrent criss-cross attention module, achieving leading performance on segmentation-based benchmarks, including Cityscapes, ADE20K, LIP, CamVid and COCO.

Compare with our original conference version [24], the following improvements are conducted: 1) We further enhance the segmentation ability of CCNet by augmenting a simple yet effective category consistent loss; 2) we propose a more generic CCNet by extending the criss-cross attention module from 2D to 3D; 3) we include more extensive experiments on the LIP, CamVid and COCO datasets to verify the effectiveness and generalization ability of our CCNet.

The rest of this paper is organized as follows. We first review related work in Section 2 and describe the architecture of our network in Section 3. In Section 4, ablation studies are given and experimental results are analyzed. Section 5 presents our conclusion and future work.

2 Related work

2.1 Semantic segmentation

The last years have seen a renewal of interest on semantic segmentation. FCN [8] is the first approach to adopt fully convolutional network for semantic segmentation. Later, FCN-based methods have made remarkable progress in image semantic segmentation. Chen *et al.* [25] and Yu *et al.* [26] removed the last two downsample layers to obtain dense prediction and utilized dilated convolutions to enlarge the receptive field. Unet [27], DeepLabv3+ [28], MSCI [29], SPGNet [30], RefineNet [31] and DFN [32] adopted encoder-decoder structures that fuse the information in low-level and high-level layers to make dense predictions. The scale-adaptive convolutions (SAC) [33] and deformable convolutional networks (DCN) [34] methods improved the standard convolutional operator to handle the deformation and various scales of objects. CRF-RNN [26] and DPN [35] used Graph model, *i.e.*, CRF, MRF, for semantic segmentation. AAF [36] used adversarial learning to capture and match the semantic relations between neighboring pixels in the label space. BiSeNet [37] was designed for real-time semantic segmentation. DenseDecoder [38] built feature-level long-range skip connections on cascaded architecture. VideoGCRF [39] used a densely-connected spatio-temporal graph for video semantic segmentation. RTA [40] proposed the region-based temporal aggregation for leveraging the temporal information in videos. In addition, some works focus on human parsing task. JPPNet [21] embed pose estimation into human parsing task. CE2P [41] proposed a simple yet effective framework for computing context embedding while preserving edges. SANet [42] used parallel branches with scale attention to handle large scale variance in human parsing. Semantic segmentation is also actively studied in the context of domain adaptation and dstillation [43], [44], [45] and weakly supervised setting [46], [47], [48], etc.

2.2 Contextual information aggregation

It is a common practice to aggregate contextual information to augment the feature representation in semantic segmentation networks. Deeplabv2 [10] proposed atrous spatial pyramid pooling (ASPP) to use different dilation convolutions to capture contextual information. DenseASPP [49] brought dense connections into ASPP to generate features with various scale. DPC [50] utilized architecture search techniques to build multi-scale architectures for semantic segmentation. Chen *et al.* [51] made use of several attention masks to fuse feature maps or prediction maps from different branches. PSPNet [11] utilized pyramid spatial pooling to aggregate contextual information. Recently, Zhao *et al.* [16] proposed the point-wise spatial attention network which uses predicted attention map to guide contextual information collection. Auto-Deeplab [52] utilized neural architecture search to search an effective context modeling. He *et al.* [53] proposed an adaptive pyramid context module for semantic segmentation. Liu *et al.* [54] utilized recurrent neural networks (RNNs) to capture long-range dependencies.

There are some works use graph models to model the contextual information. Conditional random field (CRF) [25], [40], [55], Markov random field (MRF) [35] were also utilized to capture long-range dependencies for semantic segmentation. Vaswani *et al.* [18] applied a self-attention model on machine translation. Wang *et al.* [9] proposed the non-local module to generate the huge attention map by calculating the correlation matrix between each spatial point on the feature maps, then the attention map guided dense contextual information aggregation. OCNet [56] and DANet [57] utilized Non-local module [9] to harvest the contextual information. PSA [16] learned an attention map to aggregate contextual information for each individual point adaptively and specifically. Chen *et al.* [58] proposed graph-based global reasoning networks which implements relation reasoning via graph convolution on a small graph.

CCNet vs. Non-Local vs. GCN. Here, we specifically discuss the differences among GCN [59], Non-local Network [9] and CCNet. In term of contextual information aggregation, only the center point can perceive the contextual information from all pixels by the global convolution filters in GCN [59]. In contrast, Non-local Network [9] and CCNet guarantee that a pixel at any position perceives contextual information from all pixels. Though GCN [59] alternatively decomposes the square-shape convolutional operation to horizontal and vertical linear convolutional operations which is related to CCNet, CCNet takes the criss-cross way to harvest contextual information which is more effective than the horizontal-vertical separate way. Moreover, CCNet is proposed to mimic Non-local Network [9] for obtaining dense contextual information through a more effective and efficient recurrent criss-cross attention module, in which dissimilar features get low attention weights and features with high attention weights are similar ones. GCN [59] is a conventional convolution neural network, while CCNet is a graph neural network in which each pixel in the convolutional feature map is considered as a node and the relation/context among nodes can be utilized to generate better node features.

2.3 Graph neural networks

Our work is related to deep graph neural network (GNN). Prior to graph neural networks, graphical models, such as the conditional random field (CRF) [25], [40], [55], markov random field (MRF) [35], were widely used to model the long-range dependencies for image understanding. GNNs were early studied in [15], [60], [61]. Inspired by the success of CNNs, a large number of methods adapt graph structure into CNNs. These methods could be divided into two main steams, the spectral-based approaches [62], [63], [64], [65] and the spatial-based approaches [9], [66], [67], [68]. The proposed CCNet belongs to the latter.

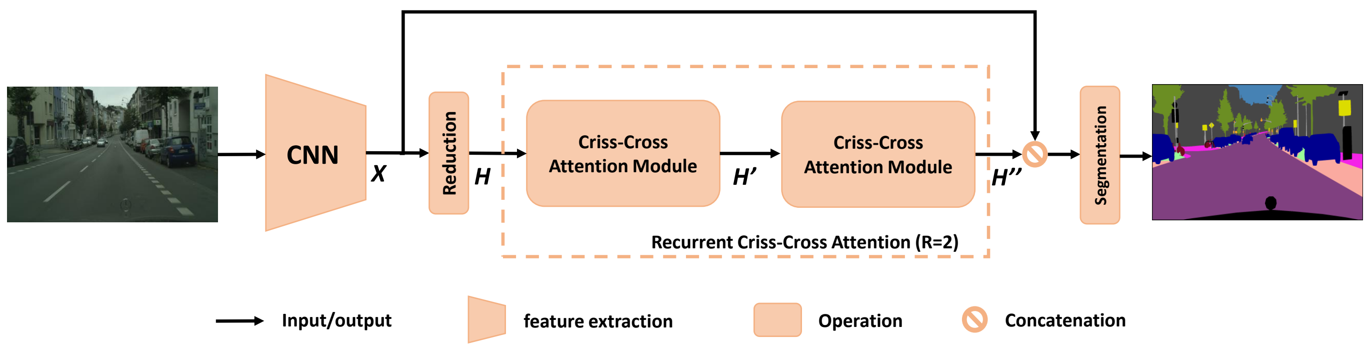
CNN
X
Reduction
H
Criss-Cross Attention Module
H'
Criss-Cross Attention Module
H''
Recurrent Criss-Cross Attention (R=2)
Segmentation
Input/output
feature extraction
Operation
Concatenation

Fig. 2. Overview of the proposed CCNet for semantic segmentation.

3 Approach

In this section, we give the details of the proposed Criss-Cross Network (CCNet) for semantic segmentation. We first present a general framework of our CCNet. Then, the 2D criss-cross attention module which captures contextual information in horizontal and vertical directions will be introduced. To capture the dense and global contextual information, we propose to adopt a recurrent operation for the criss-cross attention module. To further improve RCCA, we introduce a discriminative loss function to drive RCCA to learn category consistent features. Finally we propose the 3D criss-cross attention module for leveraging temporal and spatial contextual information simultaneously.

3.1 Network Architecture

The network architecture is given in Fig. 2. An input image is passed through a deep convolutional neural network (DCNN), which is designed in a fully convolutional fashion [10], to produce feature map $\mathbf{X}$ with the spatial size of $H \times W$. 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, leading to enlarging the width/height of the output feature map $\mathbf{X}$ to 1/8 of the input image.

Given $\mathbf{X}$, we first apply a convolutional layer to obtain the feature map $\mathbf{H}$ of dimension reduction. Then, $\mathbf{H}$ is fed into the criss-cross attention module to generate a new feature map $\mathbf{H}'$ which aggregate contextual information together for each pixel in its criss-cross path. The feature map $\mathbf{H}'$ only contains the contextual information in horizontal and vertical directions which are not powerful enough for accurate semantic segmentation. To obtain richer and denser context information, we feed the feature map $\mathbf{H}'$ into the criss-cross attention module again and output the feature map $\mathbf{H}''$. Thus, each position in $\mathbf{H}''$ actually gathers 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 predict the final segmentation result.

3.2 Criss-Cross Attention

To model full-image dependencies over local feature representations using light-weight 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 map $\mathbf{H} \in \mathbb{R}^{C \times W \times H}$, the module first applies two convolutional 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 number of channel, which is less than C for dimension reduction.

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

$$d_{i,\mathbf{u}} = \mathbf{Q_u} \mathbf{\Omega}_{i,\mathbf{u}}^{\intercal}, \tag{1}$$

where $d_{i,\mathbf{u}} \in \mathbf{D}$ is the degree of correlation between features $\mathbf{Q_u}$ and $\mathbf{\Omega}_{i,\mathbf{u}}$, $i = [1, ..., H + W - 1]$, and $\mathbf{D} \in \mathbb{R}^{(H+W-1) \times (W \times H)}$. Then, we apply a softmax layer on $\mathbf{D}$ over the channel dimension to calculate the attention map $\mathbf{A}$.

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 adaptation. At each position $\mathbf{u}$ in the spatial dimension of $\mathbf{V}$, 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 a collection of feature vectors in $\mathbf{V}$ which are in the same row or column with position u. The contextual information is collected by an **Aggregation** operation defined as follows.

$$\mathbf{H}'_{\mathbf{u}} = \sum_{i=0}^{H+W-1} \mathbf{A}_{i,\mathbf{u}} \mathbf{\Phi}_{\mathbf{i},\mathbf{u}} + \mathbf{H_u}, \tag{2}$$

where $\mathbf{H}'_{\mathbf{u}}$ is a feature vector in $\mathbf{H}' \in \mathbb{R}^{C \times W \times H}$ at position u and $\mathbf{A}_{i,\mathbf{u}}$ is a scalar value at channel i and position $\mathbf{u}$ in

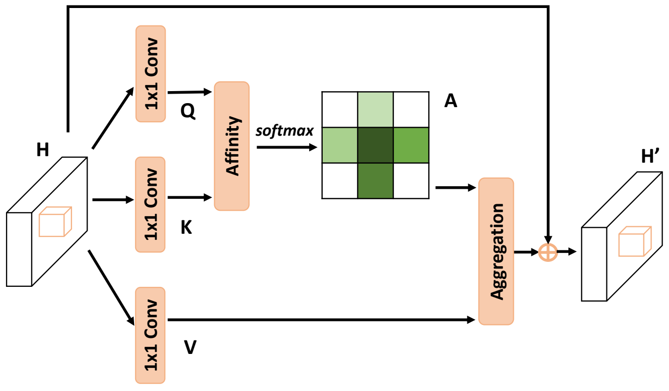

Fig. 3. The details of criss-cross attention module.

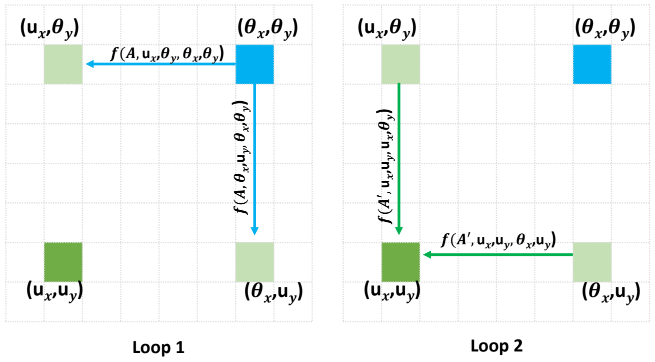
(u$_x$,θ$_y$)
(θ$_x$,θ$_y$)
f(A, u$_x$,θ$_y$, θ$_x$,θ$_y$)
f(A, θ$_x$,u$_y$, θ$_x$,θ$_y$)
(u$_x$,u$_y$)
(θ$_x$,u$_y$)
Loop 1
f(A′, u$_x$,u$_y$, u$_x$,θ$_y$)
f(A′, u$_x$,u$_y$, θ$_x$,u$_y$)
Loop 2

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

$\mathbf{A}$. The contextual information is added to local feature $\mathbf{H}$ to augment the pixel-wise representation. Therefore, it has a wide contextual view and selectively aggregates contexts according to the spatial attention map. These feature representations achieve mutual gains and are more robust for semantic segmentation.

3.3 Recurrent Criss-Cross Attention (RCCA)

Despite the criss-cross attention module can capture contextual information in horizontal and vertical directions, the connections between one pixel and its around ones that are not in the criss-cross path are still absent. To tackle this problem, we innovatively and simply introduce a RCCA operation based on the criss-cross attention. The RCCA module can be unrolled into R loops. In the first loop, the criss-cross attention takes the feature map $\mathbf{H}$ extracted from a CNN model as the input and output the feature map $\mathbf{H}'$, where $\mathbf{H}$ and $\mathbf{H}'$ are with the same shape. In the second loop, the criss-cross attention takes the feature map $\mathbf{H}'$ as the input and output the feature map $\mathbf{H}''$. As shown in Fig. 2, the RCCA module is equipped with two loops ($R = 2$) which is able to harvest full-image contextual information from all pixels to generate new features with dense and rich contextual information.

We denote $\mathbf{A}$ and $\mathbf{A}'$ 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 the identical connection. In the case of $R = 2$, the connections between any two spatial positions in the feature map built up by the RCCA module can be clearly and quantitatively described by introducing function f defined as follows.

$$\exists i \in \mathbb{R}^{H+W-1}, s.t.\ \mathbf{A}_{i,\mathbf{u}} = f(\mathbf{A}, u_x^{CC}, u_y^{CC}, u_x, u_y),$$

where $\mathbf{u}(u_x, u_y) \in \mathbb{R}^{H\times W}$ is any spatial position in $\mathbf{H}$ and $\mathbf{u}^{CC}(u_x^{CC}, u_y^{CC}) \in \mathbb{R}^{H+W-1}$ is a position in the criss-cross structure centered at $\mathbf{u}$. The function f is actually an **one-to-one mapping** from the position pair $(\mathbf{u}^{CC}, \mathbf{u}) \in \mathbb{R}^{(H+W-1)\times(H\times W)}$ in the feature map to a particular element $\mathbf{A}_{i,\mathbf{u}} \in \mathbb{R}^{(H+W-1)\times(H\times W)}$ in the attention map $\mathbf{A} \subset \mathbb{R}^{(H+W-1)\times(H\times W)}$, where $\mathbf{u}^{CC}$ maps to a particular row i in $\mathbf{A}$ and $\mathbf{u}$ maps to a particular column in $\mathbf{A}$.

With the help of function f, we can easily describe the information propagation between any position $\mathbf{u}$ in $\mathbf{H}''$ and any position $\boldsymbol{\theta}$ in $\mathbf{H}$. It is obvious that information could flow from $\boldsymbol{\theta}$ to $\mathbf{u}$ when $\boldsymbol{\theta}$ is in the criss-cross path of $\mathbf{u}$.

Then, we focus on another situation in which $\boldsymbol{\theta}(\theta_x, \theta_y)$ is NOT in the criss-cross path of $\mathbf{u}(u_x, u_y)$. To make it easier to understand, we visualize the information propagation in Fig. 4. The position (θ_x, θ_y), which is blue, firstly passes the information into the (u_x, θ_y) and (θ_x, u_y) (light green) in the loop 1. The propagation could be quantified by function f. It should be noted that these two points (u_x, θ_y) and (θ_x, u_y) are in the criss-cross path of $\mathbf{u}(u_x, u_y)$. Then, the positions (u_x, θ_y) and (θ_x, u_y) pass the information into the (u_x, u_y) (dark green) in the loop 2. Thus, the information in $\boldsymbol{\theta}(\theta_x, \theta_y)$ could eventually flow into $\mathbf{u}(u_x, u_y)$ even if $\boldsymbol{\theta}(\theta_x, \theta_y)$ is NOT in the criss-cross path of $\mathbf{u}(u_x, u_y)$.

In general, our RCCA module makes up for the deficiency of criss-cross attention that cannot obtain the dense contextual information from all pixels. Compared with criss-cross attention, the RCCA module ($R = 2$) does not bring extra parameters and can achieve better performance with the cost of a minor computation increment.

3.4 Learning Category Consistent Features

For semantic segmentation tasks, the pixels belonging to the same category should have the similar features, while the pixels from different categories should have far apart features. We name such a characteristic as category consistency. The deep features produced by RCCA have full-image context; however, the aggregated feature may have the problem of over-smoothing, which is a common issue in graph neural networks. To address this potential issue, beside the cross-entropy loss ℓ_{seg} to penalize the mismatch between the final predicted segmentation maps and ground truth, we further introduce the category consistent loss to drive RCCA module to learn category consistent features directly.

In [69], a discriminative loss function with three competing terms is proposed for instance segmentation. In particular, the three terms, denoted as $\ell_{var}, \ell_{dis}, \ell_{reg}$, are adopted to 1) penalize large distances between features with the same label for each instance, 2) penalize small distances between the mean features of different labels, and 3) draw mean features of all categories towards the origin, respectively.

Motivated by [69], we first adapt a discriminative loss for semantic segmentation rather than instance segmentation, then replace the first term with more robust one: instead of using quadratic function as the distance function to penalize mismatch all along, we design a piece-wise distance function to make the optimization more robust.

Let C be the set of classes that are present in the mini-batch images. N_c is the number of valid elements belonging to category $c \in C$. $h_i \in \mathbf{H}$ is the feature vector at spatial position i. μ_c is the mean feature of category $c \in C$ (the cluster center). φ is a piece-wise distance function. δ_v and δ_d are respectively the margins. In particular, Eq. 6 is a piece-wise distance function and the function φ_{var} will be zero, quadratic, and linear function when the distance from the center μ_c is within d_v, in range of $(\delta_v, \delta_d]$, and exceeds δ_d, respectively.

$$\ell_{var} = \frac{1}{|C|}\sum_{c\in C}\frac{1}{N_c}\sum_{i=1}^{N_c}\varphi_{var}(h_i,\mu_c), \tag{3}$$

$$\ell_{dis} = \frac{1}{|C|(|C|-1)}\sum_{\substack{c_a\in C\\ c_a\neq c_b}}\sum_{c_b\in C}\varphi_{dis}(\mu_{c_a},\mu_{c_b}), \tag{4}$$

$$\ell_{reg} = \frac{1}{|C|}\sum_{c\in C}\|\mu_c\|, \tag{5}$$

$$\varphi_{var} = \begin{cases} \|\mu_c - h_i\| - \delta_d + (\delta_d-\delta_v)^2, & \|\mu_c - h_i\| > \delta_d \\ (\|\mu_c - h_i\| - \delta_v)^2, & \delta_v < \|\mu_c - h_i\| \le \delta_d \\ 0, & \|\mu_c - h_i\| \le \delta_v \end{cases} \tag{6}$$

$$\varphi_{dis} = \begin{cases} (2\delta_d - \|\mu_{c_a} - \mu_{c_b}\|)^2, & \|\mu_{c_a} - \mu_{c_b}\| \le 2\delta_d \\ 0, & \|\mu_{c_a} - \mu_{c_b}\| > 2\delta_d \end{cases} \tag{7}$$

To reduce the computation load, we first apply a convolutional layer with 1×1 filters on the output of RCCA module for dimension reduction and then apply these three loss on the feature map with fewer channels. The final loss ℓ is weighted sum of all losses.

$$\ell = \ell_{seg} + \alpha\ell_{var} + \beta\ell_{dis} + \gamma\ell_{reg}, \tag{8}$$

where α, β and ℓ are the weight parameters. In our experiments we set $\delta_v = 0.5$, $\delta_d = 1.5$, $\alpha = \beta = 1$, $\gamma = 0.001$ and 16 as the number of channels for dimension reduction.

3.5 3D Criss-Cross Attention

To adapt our method from 2D applications to 3D dense prediction tasks, we introduce 3D Criss-Cross Attention. In general, the architecture of 3D Criss-Cross Attention is an extension the 2D version by additional collecting more contextual information from the temporal dimension. As shown in Fig. 5, given a local feature map $\mathbf{H} \in \mathbb{R}^{C\times T\times W\times H}$, where T is axial dimension (*i.e.*, temporal dimension in video data). The module firstly applies two convolutional layers with $1\times1\times1$ filters on $\mathbf{H}$ to generate two feature maps $\mathbf{Q}$ and $\mathbf{K}$, respectively, where $\{\mathbf{Q},\mathbf{K}\} \in \mathbb{R}^{C'\times T\times W\times H}$.

After obtaining the feature maps $\mathbf{Q}$ and $\mathbf{K}$, we further generate an attention map $\mathbf{A} \in \mathbb{R}^{(T+H+W-2)\times T\times W\times H}$ via

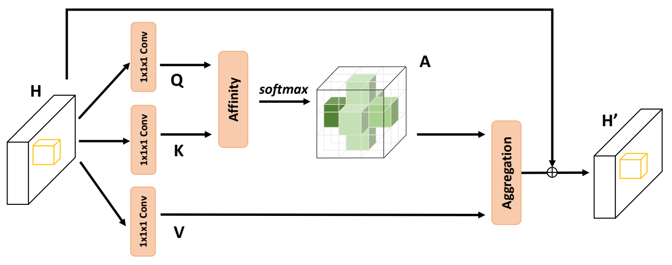

Fig. 5. The details of 3D criss-cross attention module.

the **Affinity** operation. At each position u of $\mathbf{Q}$, we can obtain a vector $\mathbf{Q_u} \in \mathbb{R}^{C'}$. u contains three coordinate values (t, x, y). We can also obtain the set $\mathbf{\Omega_u} \in \mathbb{R}^{(T+H+W-2)\times C'}$ by extracting feature vectors from $\mathbf{K}$ with at least two coordinate values equal to u. $\mathbf{\Omega_{i,u}} \in \mathbb{R}^{C'}$ is the i-th element of $\mathbf{\Omega_u}$. The **Affinity** operation is then defined as follows.

$$d_{i,u} = \mathbf{Q_u}\mathbf{\Omega_{i,u}}^{\intercal}, \tag{9}$$

where $d_{i,u} \in \mathbf{D}$ is the degree of correlation between feature $\mathbf{Q_u}$ and $\mathbf{\Omega_{i,u}}$, $i = [1, ..., (T+H+W-2)]$, $\mathbf{D} \in \mathbb{R}^{(T+H+W-2)\times T\times W\times H}$. Then, we apply a softmax layer on $\mathbf{D}$ over the first dimension to calculate the attention map $\mathbf{A}$.

Another convolutional layer with $1\times1\times1$ filters is applied on $\mathbf{H}$ to generate $\mathbf{V} \in \mathbb{R}^{C\times T\times W\times H}$ for feature adaptation. At each position u in the spatial dimension of $\mathbf{V}$, we can obtain a vector $\mathbf{V_u} \in \mathbb{R}^{C}$ and a set $\mathbf{\Phi_u} \in \mathbb{R}^{(T+H+W-2)\times C}$. The the set $\mathbf{\Phi_u}$ is a collection of feature vectors in $\mathbf{V}$ which are in the criss-cross structure centered at u. The contextual information is collected by the **Aggregation** operation:

$$\mathbf{H'_u} = \sum_{i=0}^{T+H+W-2}\mathbf{A_{i,u}}\mathbf{\Phi_{i,u}} + \mathbf{H_u}, \tag{10}$$

where $\mathbf{H'_u}$ is a feature vector in the output feature map $\mathbf{H'} \in \mathbb{R}^{C\times T\times W\times H}$ at position u. $\mathbf{A_{i,u}}$ is a scalar value at channel i and position u in $\mathbf{A}$.

4 Experiments

To evaluate the effectiveness of the CCNet, we carry out comprehensive experiments on the Cityscapes dataset [19], the ADE20K dataset [20], the COCO dataset [70], the LIP dataset [21] and the CamVid dataset [71]. Experimental results demonstrate that CCNet achieves state-of-the-art performance on Cityscapes, ADE20K and LIP. Meanwhile, CCNet can bring constant performance 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, LIP, COCO and CamVid datasets.

4.1 Datasets and Evaluation Metrics

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

- **Cityscapes** is tasked for urban segmentation. 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, respectively.
- **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 testing, respectively.
- **LIP** is a large-scale single human parsing dataset. There are 50,462 images with fine-grained annotations at pixel-level with 19 semantic human part labels and one background label. Those images are further divided into 30k/10k/10k for training, validation and testing, respectively.
- **COCO** is a very challenging dataset for instance segmentation that contains 115k images over 80 categories for training, 5k images for validation and 20k images for testing.
- **CamVid** is one of the datasets focusing on semantic segmentation for autonomous driving scenarios. It is composed of 701 densely annotated images with size 720×960 from five video sequences.

4.2 Implementation Details

Network Structure For semantic segmentation, we choose the ImageNet pre-trained ResNet-101 [23] as our backbone network, remove its last two down-sampling operations, and employ dilated convolutions in the subsequent convolutional layers following the previous work [25], resulting in the output stride as 8. For human parsing, we choose CE2P [41] as our baseline and replace the Context Embedding module with RCCA. For instance segmentation, we choose Mask-RCNN [72] as our baseline. For video semantic segmentation, we also choose Cityscapes pre-trained ResNet-101 [23] as our backbone network with 3D RCCA.

Training settings SGD with mini-batch is used for training. For semantic segmentation, the initial learning rate is 1e-2 for Cityscapes and ADE20K. Following the prior works [10], [14], 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), then randomly cropping out 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 a length randomly chosen from the set {300, 375, 450, 525, 600}. For human parsing, the model are trained and tested with the input size of 473×473. For instance segmentation, we take the same training settings as that of Mask-RCNN [72]. For video semantic segmentation, we sample 5 temporally ordered frames from a training video as training data and the input size is 504×504.

TABLE 1
Comparison with state-of-the-arts on Cityscapes (test).

Method	Backbone	mIOU(%)
Performance on val set		
DeepLabv3 [12]	ResNet-101	79.3
DeepLabv3+ [28]	Xception-65	79.1
DPC [50] †	Xception-71	80.8
CCNet	ResNet-101	80.5
Performance on test set		
DeepLab-v2 [10]	ResNet-101	70.4
RefineNet [31] ‡	ResNet-101	73.6
SAC [33] ‡	ResNet-101	78.1
GCN [59] ‡	ResNet-101	76.9
DUC [73] ‡	ResNet-101	77.6
ResNet-38 [74]	WiderResnet-38	78.4
PSPNet [11]	ResNet-101	78.4
BiSeNet [37] ‡	ResNet-101	78.9
AAF [36]	ResNet-101	79.1
PSANet [16] ‡	ResNet-101	80.1
DFN [32] ‡	ResNet-101	79.3
DenseASPP [49] ‡	DenseNet-161	80.6
CCNet ‡	ResNet-101	**81.9**

† use extra COCO dataset for training.
‡ train with both the train-fine and val-fine datasets.

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 are summarized in Tab. 1. For val set, we provide these results for reference and emphasize that these results should not be simply compared with our method, since these methods are trained on different (even larger) training sets or different basic network. Among these approaches, Deeplabv3 [12] adopts multi-scale testing strategy. Deeplabv3+ [28] and DPC [50] both use a more stronger backbone (*i.e.*, Xception-65 & 71 *vs.* ResNet-101). In addition, DPC [50] makes use of additional dataset, *i.e.*, COCO, for pre-training beyond the training set of Cityscapes. The results show that the proposed CCNet with single-scale testing still achieve comparable performance without bells and whistles.

Additionally, we also train the best learned CCNet with ResNet-101 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 [10], [11], [16], [31], [32], [33], [36], [37], [59], [73] adopt the same backbone as ours and the others [49], [74] utilize stronger backbones. From Tab. 1, it can be observed that our CCNet substantially outperforms all the previous state-of-the-arts on test set. Among the approaches, PSANet [16] is the 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. Even with lower computation cost and memory usage, our method still achieves better performance.

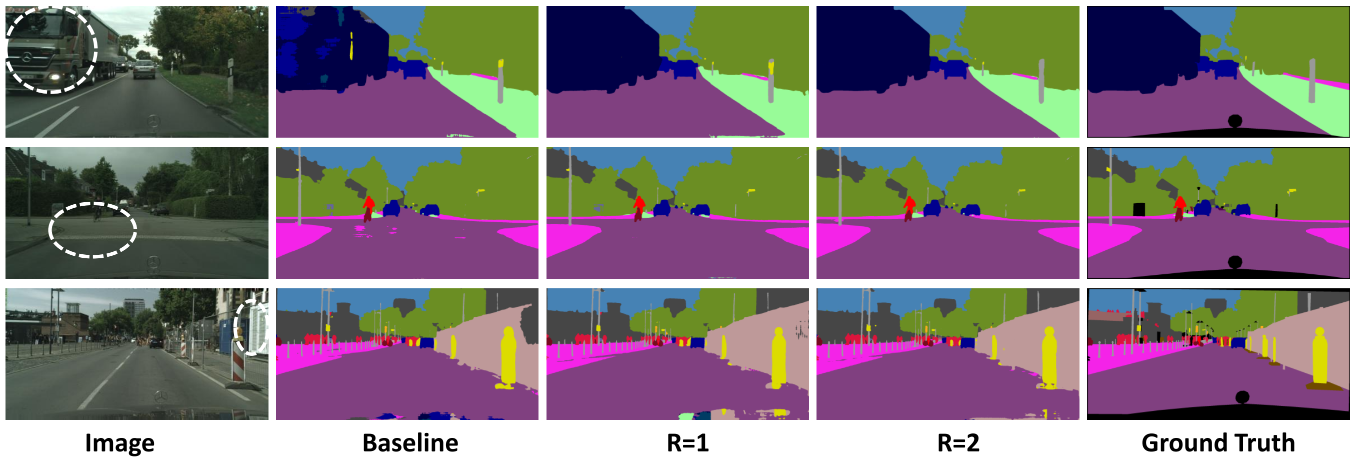

Fig. 6. Visualization results of RCCA with different loops on Cityscapes validation set.

TABLE 2
Performance on Cityscapes (val) for different number of loops in RCCA. FLOPs and memory increment 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 3
Performance on Cityscapes (val) for different kinds of category consistent loss.

Function Type	Successes	Mean mIOU(%)
Quadratic function	6/10	79.2
Piece-wise function	9/10	79.3

4.3.2 Ablation studies

To verify the rationality of the CCNet, we conduct extensive ablation experiments on the validation set of Cityscapes with different settings for CCNet.

The effect of the RCCA module Tab. 2 shows the performance on the Cityscapes validation set by adopting different number of loop in RCCA. All experiments are conducted using ResNet-101 as the backbone. Besides, the input size of training images is 769×769 and the size of the input feature map $\mathbf{H}$ of RCCA is 97×97. Our baseline network is the ResNet-based FCN with dilated convolutional module incorporated at stage 4 and 5, *i.e.*, dilation rates 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 observe that adding a criss-cross attention module into the baseline, donated as $R = 1$, improves the performance by 2.9%, which can effectively demonstrates the significance of criss-cross attention. Furthermore, increasing the number of loops from 1 to 2 can further 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 the number of loops, the FLOPs and usage of GPU memory keep increasing. These results prove that the proposed criss-cross attention can significantly improve the performance by capturing contextual information in horizontal and vertical direction. In addition, the proposed RCCA is effective in capturing the dense and global contextual information, which can finally 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.

To further validate the effectiveness of the criss-cross module, we provide the qualitative comparisons in Fig. 6. We leverage the *white circles* to indicate those challenging regions that are easily to be misclassified. It can be seen that these challenging regions are progressively corrected with the increasing the number of loops, which can well prove the effectiveness of dense contextual information aggregation for semantic segmentation.

The effect of the category consistent loss Tab. 4 also shows the performance on the Cityscapes validation set by adopting the proposed category consistent loss. The category consistent loss is donated as "CCL" in the table. As we can see, adopting the category consistent loss could stably bring $\tilde{0}.7\%$ mIoU gain with both Resnet-101 and Resnet-50, which prove the effectiveness of the proposed category consistent loss for semantic segmentation. To prove that the proposed piece-wise function is more robust than the original one, we conduct 10 times of the training processes using ResNet-50 for each kind of loss function. The training is deemed to fail when the loss value is NaN, thus we can calculate the success rate (number of successful training / total number of training). The experimental results in Table 3 demonstrate that using the piece-wise function has higher training success rate than using the original one. Besides, using the piece-wise function could achieve slightly better performance than a single quadratic function. Because we

TABLE 4
Comparison of context aggregation approaches on Cityscapes (val).

Method	mIOU(%)
ResNet101-Baseline	75.1
ResNet101+GCN	78.1
ResNet101+PSP	78.5
ResNet101+ASPP	78.9
ResNet101+NL	79.1
ResNet101+RCCA(R=2)	79.8
ResNet101+RCCA(R=2)+CCL	**80.5**
ResNet50-Baseline	73.3
ResNet50+GCN	76.2
ResNet50+PSP	76.4
ResNet50+ASPP	77.1
ResNet50+NL	77.3
ResNet50+HV	77.3
ResNet50+HV&VH	77.8
ResNet50+RCCA(R=2)	78.5
ResNet50+RCCA(R=2)+CCL	**79.3**

relax the punishment in the Eq. 6 to reduce the numerical values and gradients especially when the distance from the center exceeds δ_d. This relaxation makes the optimization much more stable.

Comparison of other 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 backbone networks.

Specifically, the baselines of context aggregation mainly include: 1) Peng *et al.* [59] utilized global convolution filters for contextual information aggregation, donated as "+GCN". 2) Zhao *et al.* [11] proposed Pyramid pooling which is the simple and effective way to capture global contextual information, donated as "+PSP"; 3) Chen *et al.* [12] used different dilation convolutions to harvest pixel-wise contextual information at the different range, donated as "+ASPP"; 4) Wang *et al.* [9] introduced non-local network for context aggregation, donated as "+NL".

In Tab. 4, both "+NL" and "+RCCA" achieve better performance compared with the other context aggregation approaches, which demonstrates the importance of capturing full-image contextual information. More interestingly, our method achieves better performance than "+NL". This reason may be attributed to the sequentially recurrent operation of criss-cross attention. Concretely, "+NL" generates an attention map directly from the feature which has limit receptive field and short-range dependencies. In contrast, 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-range dependencies has already been embedded.

To prove the effectiveness of attention with criss-cross shape, we compare criss-cross shape with other shapes in Tab. 4. "+HV" means stacking horizontal attention and vertical attention. "+HV&VH" means summing up features of two parallel branches, i.e. "HV" and "VH".

TABLE 5
Comparison of Non-local module and RCCA. FLOPs and memory increment 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
+NL(R=2)	216	2820	78.7
+RCCA(R=2)	16.5	127	78.5

We further explore the amount of computation and memory footprint of RCCA. As shown in Tab. 5, compared with "+NL" method, the proposed "+RCCA" requires 11× less GPU memory usage and significantly reduces FLOPs by about 85% of non-local block in computing full-image dependencies, which shows that CCNet is an efficient way to capture full-image contextual information in the least amount of computation and memory footprint. To further prove the effectiveness of the recurrent operation, we also run non-local module in the recurrent way, donated as "+NL(R=2)". As we can seen, the recurrent operation can bring more than 1 point gain. Because the recurrent operation leads 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-range dependencies has already been embedded. However, compared with "+RCCA", "+NL(R=2)" needs huge GPU memory usage, which limits the use of self-attention.

Visualization of Attention Map To get a deeper understanding of our RCCA, we visualize the learned attention masks as shown in Fig. 7. For each input image, we select one point (cross in green) and show its corresponding attention maps when $R = 1$ and $R = 2$ in columns 2 and 3, respectively. It can be observed that only contextual information from the criss-cross path of the target point is captured when $R = 1$. By adopting one more criss-cross module, *i.e.*, $R = 2$, 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 full-image dependencies.

4.4 Experiments on ADE20K

In this subsection, we conduct experiments on the AED20K dataset, which is a very challenging scene parsing dataset. As shown in Tab. 6, CCNet with CCL achieves the state-of-the-art performance of 45.76%, outperforms the previous state-of-the-art methods by more than 1.1% and also outperforms the conference version CCNet by 0.5%. Some successful segmentation results are given in Fig 8. Among the approaches, most of methods [11], [14], [16], [33], [75], [76] adopt the ResNet-101 as backbone and RefineNet [31] adopts a more powerful network, *i.e.*, ResNet-152, as the backbone. EncNet [14] 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 full-image dependencies and achieve better performance.

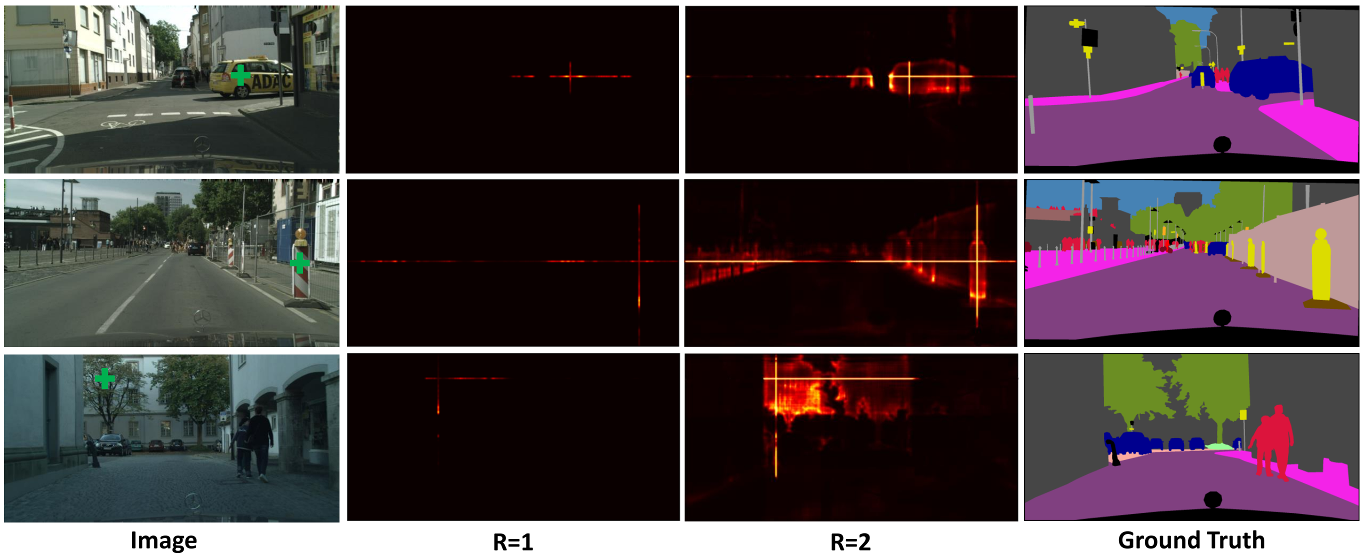

Fig. 7. Visualization of attention module on Cityscapes validation set. The left column is the input images, the 2 and 3 columns are pixel-wise attention maps when $R = 1$ and $R = 2$ in RCCA.

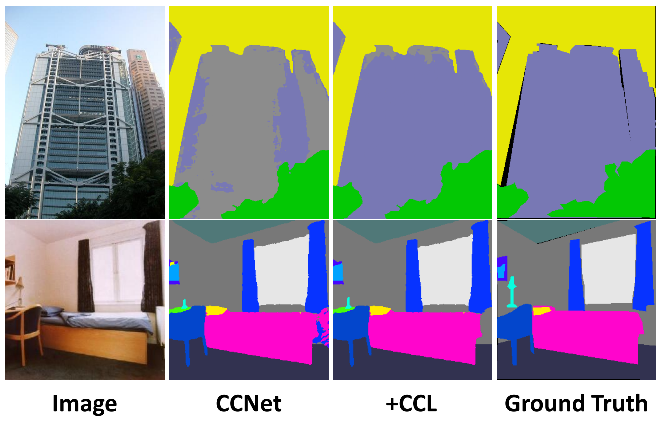

Fig. 8. Visualized examples on ADE20K val set with/without category consistent loss (CCL).

4.5 Experiments on LIP

In this subsection, we conduct experiments on the LIP dataset, which is a very challenging human parsing dataset. The framework of CE2P [41] is utilized, with ImageNet pre-trained ResNet-101 as bockbone and using RCCA (R=2) rather than PSP [11] as context embedding module. The category consistent loss is used to boost the performance. The hyper-parameter setting strictly follows that in the CE2P [41]. Among the approaches, Deeplab (VGG-16) [25], Attention [51] and SAN [42] adopt the VGG-16 as backbone and Deeplab (ResNet-101) [10], JPPNet [21], CE2P [41] and CCNet adopt ResNet-101 as the backbone. As shown in Tab. 7, CCNet achieves the state-of-the-art performance of 55.47%, outperforms the previous state-of-the-art methods by more than 2.3%. This significant improvement demonstrates the effectiveness of proposed method on human parsing task. Fig. 9 shows some visualized segmentation results. The top two rows show some successful segmentation results It shows our method can produce accurate segmentation even for complicated poses. The third row shows a failure segmentation result where the "skirt" is misclassified as "pants". But it's difficult to recognize even for humans.

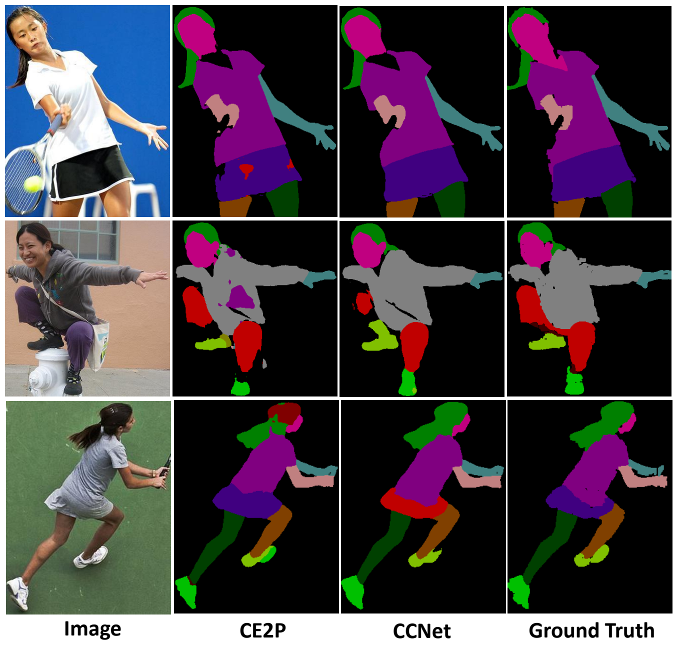

Fig. 9. Visualized examples for human parsing result on LIP val set.

4.6 Experiments on COCO

To further demonstrate the generality of CCNet, we conduct the instance segmentation task on COCO [70] using the competitive Mask R-CNN model [72] as the baseline. Following [9], 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

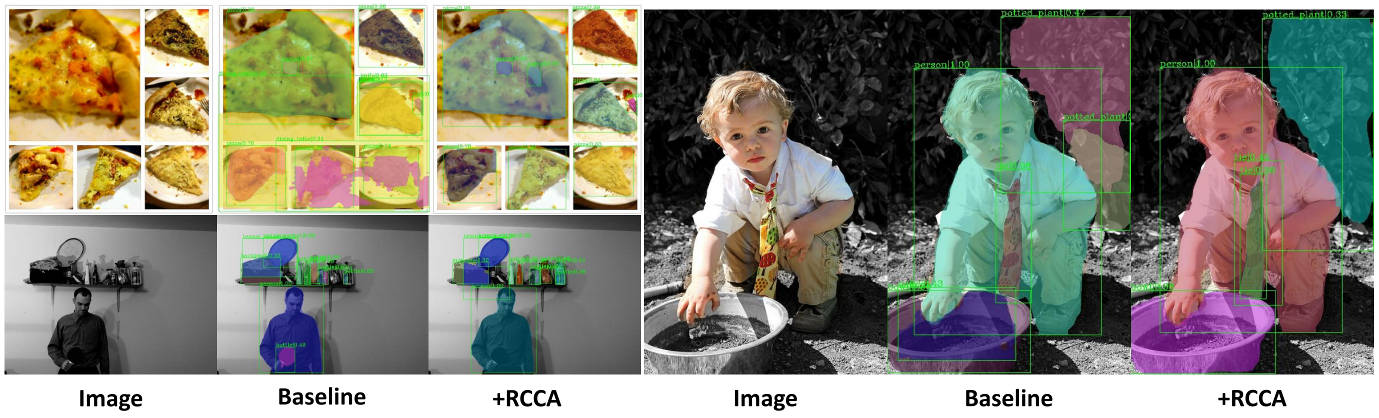

Fig. 10. Visualized examples for instance segmentation result on COCO val set.

pre-training. We use the official implementation[1] with end-to-end joint training whose performance is almost the same as the baseline reported in [9]. For fair comparison, we do not use the category consistent loss in our method. We report the results in terms of box AP and mask AP in Tab. 8 on COCO. The results demonstrate that our method substantially outperforms the baseline in all metrics. Some segmentation results for comparing baseline with "+RCCA" are given in Fig 10. Meanwhile, the network with "+RCCA" also achieves the better performance than the network with one non-local block "+NL".

4.7 Experiments on CamVid

To further demonstrate the effectiveness of 3D-RCCA, we carry out the experiments on CamVid [71], which is one of the first datasets focusing on video semantic segmentation for driving scenarios. We follow the standard protocol proposed in [77] to split the dataset into 367 training, 101 validation and 233 test images. For fair comparison, we only report single-scale evaluation scores. As can be seen in Tab. 9, we achieve an mIoU of 79.1%, outperforming all other methods by a large margin.

To demonstrate the effectiveness of our proposed techniques, we perform training under the same settings with the different length of input frames. We apply the CNNs on each frame for extracting features and then concatenate and reshape them to satisfy the required shape of 3D Criss-Coss Attention module. We use the $R = 3$ for collecting dense spatial and temporal contextual information. Here, to make a training sample, we try two kinds of length (T) of input frames. For $T = 1$, we randomly sample 1 frame from a training video, donated as "CCNet3D ($T = 1$)". For $T = 5$, we sample 5 temporally ordered frames from a training video, donated as "CCNet3D ($T = 5$)". As can be seen in Tab. 9, "CCNet3D ($T = 5$)" outperforms "CCNet3D ($T = 1$)" by 1.2%.

5 Conclusion and Future Work

In this paper, we have presented a Criss-Cross Network (CCNet) for deep learning based dense prediction tasks, which adaptively captures contextual information on the criss-cross path. To obtain dense contextual information, we introduce RCCA which aggregates contextual information

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

TABLE 6
Comparison with state-of-the-arts on ADE20K (val).

Method	Backbone	mIOU(%)
RefineNet [31]	ResNet-152	40.70
SAC [33]	ResNet-101	44.30
PSPNet [11]	ResNet-101	43.29
PSANet [16]	ResNet-101	43.77
DSSPN [75]	ResNet-101	43.68
UperNet [76]	ResNet-101	42.66
EncNet [14]	ResNet-101	44.65
CCNet	ResNet-101	**45.76**

TABLE 7
Comparison with state-of-the-arts on LIP (val).

Method	pixel acc	mean acc	mIoU
DeepLab (VGG-16) [10]	82.66	51.64	41.64
Attention [51]	83.43	54.39	42.92
SAN [42]	84.22	55.09	44.81
DeepLab (ResNet-101) [10]	84.09	55.63	44.80
JPPNet [21]	86.39	62.32	51.37
CE2P [41]	87.37	63.20	53.10
CCNet	**88.01**	**63.91**	**55.47**

TABLE 8
Comparison on COCO (val).

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**

TABLE 9
Results on the CamVid test set.

Method	Bockbone	mIoU (%)
SegNet [77]	VGG16	60.1
RTA [40]	VGG16	62.5
Dilate8 [26]	Dilate	65.3
BiSeNet [37]	ResNet18	68.7
PSPNet [11]	ResNet50	69.1
DenseDecoder [38]	ResNeXt101	70.9
VideoGCRF‡ [39]	ResNet101	75.2
CCNet3D (T=1) ‡	ResNet101	77.9
CCNet3D (T=5) ‡	ResNet101	**79.1**

‡ the initialized model is pre-trained on Cityscapes.

from all pixels. The experiments demonstrate that RCCA captures full-image contextual information in less computation cost and less memory cost. Besides, to learn discriminative features, we introduce the category consistent loss. Our CCNet achieves outstanding performance consistently on several semantic segmentation datasets, *i.e.*, Cityscapes, ADE20K, LIP, CamVid and instance segmentation dataset, *i.e.*, COCO. The source codes of CCNet are released to facilitate related research and applications.

Acknowledgements

This work was in part supported by NSFC (No. 61733007 and No. 61876212), ARC DECRA DE190101315, ARC DP200100938, HUST-Horizon Computer Vision Research Center, and IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network.

References

[1] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *ITSC*, 2013, pp. 1693–1700. 1

[2] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997. 1

[3] M. Evening, *Adobe Photoshop CS3 for photographers: a professional image editor's guide to the creative use of Photoshop for the Macintosh and PC*. Focal press, 2012. 1

[4] Y. Song, Z. Huang, C. Shen, H. Shi, and D. A. Lange, "Deep learning-based automated image segmentation for concrete petrographic analysis," *Cement and Concrete Research*, vol. 135, p. 106118, 2020. 1

[5] Z. Zheng, Y. Zhong, J. Wang, and A. Ma, "Foreground-aware relation network for geospatial object segmentation in high spatial resolution remote sensing imagery," in *CVPR*, 2020. 1

[6] M. T. Chiu, X. Xu, Y. Wei, Z. Huang, A. G. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose *et al.*, "Agriculture-vision: A large aerial image database for agricultural pattern analysis," in *CVPR*, 2020, pp. 2828–2838. 1

[7] M. Tik Chiu, X. Xu, K. Wang, J. Hobbs, N. Hovakimyan, S. H. Huang, Thomas S *et al.*, "The 1st agriculture-vision challenge: Methods and results," in *CVPR Workshops*, 2020, pp. 48–49. 1

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440. 1, 3

[9] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018, pp. 7794–7803. 1, 2, 3, 9, 10, 11

[10] 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 TPAMI*, vol. 40, no. 4, pp. 834–848, 2018. 1, 2, 3, 4, 7, 8, 10, 11

[11] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890. 2, 3, 7, 8, 9, 10, 11, 12

[12] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. 2, 7, 8, 9

[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 *CVPR*, June 2018. 2

[14] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *CVPR*, 2018. 2, 7, 9, 11

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE TNN*, vol. 20, no. 1, pp. 61–80, 2008. 2, 3

[16] 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 *ECCV*, 2018, pp. 270–286. 2, 3, 7, 8, 9, 11

[17] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," 2016. 2

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008. 2, 3

[19] 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 *CVPR*, 2016, pp. 3213–3223. 2, 6

[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017. 2, 6

[21] X. Liang, K. Gong, X. Shen, and L. Lin, "Look into person: Joint body parsing & pose estimation network and a new benchmark," *IEEE TPAMI*, vol. 41, no. 4, pp. 871–885, 2018. 2, 3, 6, 10, 11

[22] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009. 2

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778. 2, 7

[24] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *ICCV*, 2019. 2

[25] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *ICLR*, 2015. 3, 7, 10

[26] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *ICLR*, 2016. 3, 12

[27] 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*, 2015, pp. 234–241. 3

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *ECCV*, 2018. 3, 7, 8

[29] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *ECCV*, 2018, pp. 603–619. 3

[30] B. Cheng, L.-C. Chen, Y. Wei, Y. Zhu, Z. Huang, J. Xiong, T. Huang, W.-M. Hwu, and H. Shi, "Spgnet: Semantic prediction guidance for scene parsing," in *ICCV*, 2019. 3

[31] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." in *CVPR*, 2017. 3, 7, 8, 9, 11

[32] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," *CVPR*, 2018. 3, 7, 8

[33] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *ICCV*, 2017, pp. 2031–2039. 3, 7, 8, 9, 11

[34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *ICCV*, 2017. 3

[35] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *ICCV*, 2015, pp. 1377–1385. 3

[36] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu, "Adaptive affinity field for semantic segmentation," *ECCV*, 2018. 3, 7, 8

[37] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," *ECCV*, 2018. 3, 7, 8, 12

[38] P. Bilinski and V. Prisacariu, "Dense decoder shortcut connections for single-pass semantic segmentation," in *CVPR*, 2018, pp. 6596–6605. 3, 12

[39] S. Chandra, C. Couprie, and I. Kokkinos, "Deep spatio-temporal random fields for efficient video segmentation," in *CVPR*, 2018, pp. 8915–8924. 3, 12

[40] P.-Y. Huang, W.-T. Hsu, C.-Y. Chiu, T.-F. Wu, and M. Sun, "Efficient uncertainty estimation for semantic segmentation in videos," in *ECCV*, 2018, pp. 520–535. 3, 12

[41] T. Ruan, T. Liu, Z. Huang, Y. Wei, S. Wei, and Y. Zhao, "Devil in the details: Towards accurate single and multiple human parsing," in *AAAI*, vol. 33, 2019, pp. 4814–4821. 3, 7, 10, 11

[42] Z. Huang, C. Wang, X. Wang, W. Liu, and J. Wang, "Semantic image segmentation by scale-adaptive networks," *IEEE TIP*, 2019. 3, 10, 11

[43] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W.-m. Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *CVPR*, 2020, pp. 12 635–12 644. 3

[44] Z. Wang, Y. Wei, R. Feris, J. Xiong, W.-M. Hwu, T. S. Huang, and H. Shi, "Alleviating semantic-level shift: A semi-supervised domain adaptation method for semantic segmentation," in *CVPR Workshops*, 2020, pp. 936–937. 3

[45] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. W. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in *CVPR*, 2019, pp. 2869–2878. 3

[46] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *CVPR*, 2018, pp. 7014–7023. 3

[47] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *CVPR*, 2018, pp. 7268–7277. 3

[48] R. Qian, Y. Wei, H. Shi, J. Li, J. Liu, and T. Huang, "Weakly supervised scene parsing with point-based distance metric learning," in *AAAI*, vol. 33, 2019, pp. 8843–8850. 3

[49] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *CVPR*, 2018, pp. 3684–3692. 3, 7, 8

[50] 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," *NeurIPS*, 2018. 3, 7, 8

[51] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *CVPR*, 2016, pp. 3640–3649. 3, 10, 11

[52] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *CVPR*, 2019, pp. 82–92. 3

[53] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *CVPR*, 2019, pp. 7519–7528. 3

[54] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, "Learning affinity via spatial propagation networks," in *NeurIPS*, 2017, pp. 1520–1530. 3

[55] 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 *ICCV*, 2015, pp. 1529–1537. 3

[56] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018. 3

[57] J. Fu, J. Liu, H. Tian, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," *CVPR*, 2019. 3

[58] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *CVPR*, 2019, pp. 433–442. 3

[59] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel mattersimprove semantic segmentation by global convolutional network," in *CVPR*, 2017, pp. 1743–1751. 3, 7, 8, 9

[60] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE TNN*, vol. 8, no. 3, pp. 714–735, 1997. 3

[61] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *IJCNN*, vol. 2, 2005, pp. 729–734. 3

[62] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015. 3

[63] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NeurIPS*, 2016, pp. 3844–3852. 3

[64] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. 3

[65] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE TSP*, vol. 67, no. 1, pp. 97–109, 2018. 3

[66] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *NeurIPS*, 2016, pp. 1993–2001. 3

[67] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *ICML*, 2016, pp. 2014–2023. 3

[68] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017, pp. 1263–1272. 3

[69] B. De Brabandere, D. Neven, and L. Van Gool, "Semantic instance segmentation with a discriminative loss function," *arXiv preprint arXiv:1708.02551*, 2017. 5, 6

[70] 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 *ECCV*, 2014, pp. 740–755. 6, 10

[71] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008, pp. 44–57. 6, 11

[72] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988. 7, 10

[73] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *WACV*, 2018, pp. 1451–1460. 7, 8

[74] 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. 7, 8

[75] X. Liang, H. Zhou, and E. Xing, "Dynamic-structured semantic propagation network," in *CVPR*, 2018, pp. 752–761. 9, 11

[76] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," *ECCV*, 2018. 9, 11

[77] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE TPAMI*, vol. 39, no. 12, pp. 2481–2495, 2017. 11, 12