2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Boundary-aware Graph Convolution for Semantic Segmentation

Hanzhe Hu, Jinshi Cui and Hongbin Zha
School of Electronics Engineering and Computer Science, Peking University
huhz@pku.edu.cn, cjs@cis.pku.edu.cn, zha@cis.pku.edu.cn

***Abstract*—Recent works have made great progress in semantic segmentation by exploiting contextual information in a local or global manner with dilated convolutions, pyramid pooling or self-attention mechanism. However, few works have focused on harvesting boundary information to improve the segmentation performance. In order to enhance the feature similarity within the object and keep discrimination from other objects, we propose a boundary-aware graph convolution (BGC) module to propagate features within the object. The graph reasoning is performed among pixels of the same object apart from the boundary pixels. Based on the proposed BGC module, we further introduce the Boundary-aware Graph Convolution Network(BGCNet), which consists of two main components including a basic segmentation network and the BGC module, forming a coarse-to-fine paradigm. Specifically, the BGC module takes the coarse segmentation feature map as node features and boundary prediction to guide graph construction. After graph convolution, the reasoned feature and the input feature are fused together to get the refined feature, producing the refined segmentation result. We conduct extensive experiments on three popular semantic segmentation benchmarks including Cityscapes, PASCAL VOC 2012 and COCO Stuff, and achieve state-of-the-art performance on all three benchmarks.**

I. Introduction

Semantic Segmentation is a fundamental and challenging problem in computer vision, which aims to assign a category label to each pixel in an image. It has been widely applied to many scenarios, such as autonomous driving, scene understanding and image editing.

Recent state-of-the-art semantic segmentation methods based on the fully convolutional network(FCN) [1] have made great progress. To capture the long-range contextual information, the atrous spatial pyramid pooling(ASPP) module in DeepLabv3 [2] aggregates spatial regularly sampled pixels at different dilated rates and the pyramid pooling module in PSPNet [3] partitions the feature maps into multiple regions before pooling. More comprehensively, PSANet [4] was proposed to generate dense and pixel-wise contextual information, which learns to aggregate information via a predicted attention map. Non-local Network [5] adopts self-attention mechanism, which enables every pixel to receive information from every other pixels in the image, resulting in a much complete pixel-wise representation.

Since the key of segmentation is to segment a scene image to semantic regions, it is useful to enhance the similarity of the same object while keeping the discrimination of other objects. In other words, feature propagation only occurs within the object region, which is beneficial for feature learning in improving feature similarity, as shown in Fig.1. In order to realize this objective, we make use of boundary information to control the information propagation flow during feature learning. In this way, feature similarity and discrimination are both enhanced within and between objects, respectively. Since graph convolution is good at passing information with the design of adjacency matrix and proves to be a good reasoning method, we utilize graph convolution to control the flow during learning process. Aiming to address the above concerns, we propose a boundary-aware graph convolution(BGC) module and further the BGCNet, which can effectively enhance the feature similarity within an object and at the same time increase the difference between objects. The main design of our proposed module contains two parts: boundary detection and graph convolution.

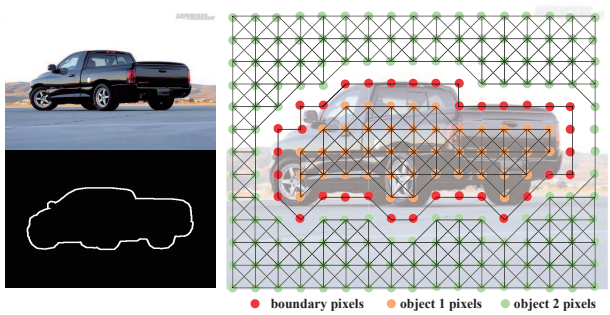

Fig. 1. Illustration of our proposed feature propagation design. The connections only exists within the same object(pixels of object 1 are not connected to pixels of object 2), which helps to enhance the feature similarity of the same object and keep discrimination of others.

First, boundary detection, which is an important issue in semantic segmentation, however, attracts few attention since it brings little improvement to the final performance due to the small portion of pixels of boundaries. Though previous works design explicit models to detect the boundaries, we intend to obtain the preliminary segmentation result and boundary map simultaneously. With regard to this purpose, we first generate the boundary label from the existing class labels given in the segmentation datasets and define it as an additional class for learning. In this way, without bringing much computation overhead, we can achieve the coarse boundary prediction from a basic semantic segmentation network, which can be further

utilized for feature propagation control.

Second, graph convolution, which requires the adjacency matrix to control the information flow, has been popularly used as a reasoning method. Methods based on convolutional networks are very common in semantic segmentation problems these days. However, on the one hand, regular convolution operation takes neighbors around a pixel equally into consideration and cannot block the information flow across boundaries, hence increasing the feature ambiguity between semantic objects. While on the other hand, with specifically-designed adjacency matrix, graph convolution can effectively control the feature propagation manner. Concisely, we treat the nodes of downsampled feature map as graph nodes, and add edges between nodes, producing a sparsely-connected graph and perform graph convolution for feature learning.

Finally, based on the learned boundary and boundary-aware graph convolution, we propose the boundary-aware graph convolution network(BGCNet). The overall framework is shown in Fig.2, which follows a coarse-to-fine paradigm. The input image is first fed into a basic network which can be any segmentation network, outputting the coarse feature map together with its corresponding coarse prediction result including boundary prediction. Subsequently, the BGC module takes the coarse feature as input and constructs the graph under the instruction of predicted boundary. After graph convolution, the reasoned feature is concatenated with the input feature to obtain the refined feature. Finally, the final refined segmentation result is achieved from the refined feature.

The main contributions of this paper are summarized as follows:

- We utilize a coarse-to-fine framework and obtain boundary prediction from coarse feature by learning it as one of the semantic categories with little increase on computation overhead.
- We propose the boundary-aware graph convolution module to enhance the feature similarity within one object and keep discrimination from others.
- We conduct extensive experiments on several public datasets, and obtain state-of-the-art performances on the Cityscapes [6], PASCAL VOC 2012 [7] and COCO Stuff [8] datasets.

II. Related Work

A. Semantic Segmentation.

Benefiting from the success of deep neural networks [9]–[11], semantic segmentation has achieved great progress. FCN [1] is the first approach to adopt fully convolutional network for semantic segmentation. Later, many FCN-baed works are proposed, such as UNet [12], SegNet [13], RefineNet [14], PSPNet [3], DeepLab series [2], [15]–[17]. Chen *et al.* [16] and Yu *et al.* [18] removed the last two downsample layers to obtain a dense prediction and utilized dilated convolutions to enlarge the receptive field. In our model, we also adopt the above paradigm to get a better feature map and hence, improve the performance of the model.

B. Graph Convolution.

Graph-based methods have been very popular these days and shown to be an efficient way of relation reasoning. CRF [19]is proposed based on the graph model for image segmentation and works as an effective postprocessing method in DeepLab [16]. Recently, Graph Convolution Networks(GCN) [20] are proposed for semi-supervised classification, and Wang *et al.* [21] use GCN to capture relations between objects in video recognition tasks. Later, a few works based on GCN have been proposed onto the semantic segmentation problem, including [22], [23],which all similarly model the relations between regions of the image rather than individual pixels. Concretely, clusters of pixels are defined as the vertices of the graph, hence graph reasoning is performed in the intermediate space projected from the original feature space to reduce computation cost, thus losing some ratio of information. Different from these recent GCN-based methods, we perform graph convolution in a boundary-aware manner, where boundary works as obstructing information flow between objects hence enhancing the feature similarity inside one object.

C. Boundary Detection

Boundary detection is a fundamental task in computer vision. Recently, the bloom of CNNs has greatly improved the performance of boundary detection [24]–[27]. [27] utilizes features from intermediate layers of CNNs for boundary detection while [25] proposes multi-stage fully convolutional networks for boundary detection. These works focus on optimizing the accuracy of boundary detection instead of using it for higher-level vision tasks. However, boundary information can be well harnessed in segmentation problems. [24], [28]–[30] employ edges to improve segmentation performance with additional branch for boundary detection. While [31], [32] treat boundary pixels as hard samples and assign additional attention to these hard pixels such as more loss functions or MLP operations. Different from works mentioned above, we embed boundary prediction task into semantic segmentation task, where the two tasks are learned together and hence benefit each other, producing the explicit boundary map. Moreover, this process brings little extra computation cost compared with two branches method from previous works, and we can use the predicted boundary map for further feature propagation design.

III. Approach

A. Preliminaries

Graph Convolution. Given an input feature $X \in \mathbb{R}^{N \times D}$, where N is the number of nodes in the feature map and D is the feature dimension, we can build a feature graph G from this input feature. Specifically, the graph G can be formulated as $G = \{V, \varepsilon, A\}$ with V as its nodes, ε as its edges and A as its adjacency matrix. Normally, the adjacency matrix A is a binary matrix, in practice, we try many ways to construct the graph, including top-k binary matrix or dynamic learnable matrix, and further design a novel boundary-aware sampling method

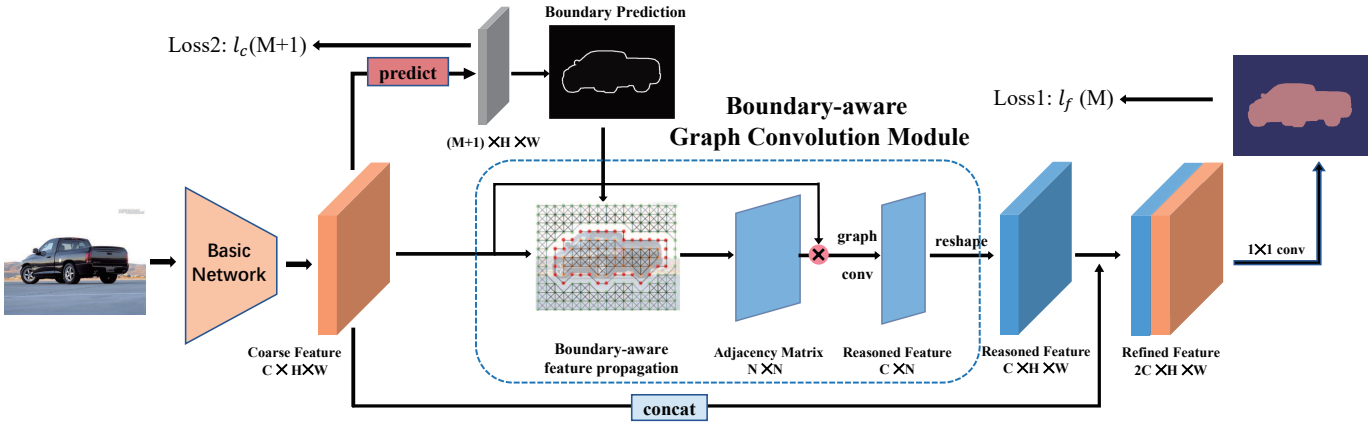

Fig. 2. An overview of boundary-aware graph convolution network. Given an input image, we first feed it into the basic network to get the high-level feature map(coarse feature) and the corresponding coarse segmentation result including the binary boundary map. With the coarse feature and boundary map, the BGC module performs graph reasoning along nodes of the feature map, producing the reasoned feature which is to be fused with the coarse feature to produce the refined feature. Specifically, loss 1 is the conventional segmentation loss of M classes defined in the dataset, while loss 2 is supervised with the new groundtruth of M+1 classes with an additional boundary class generated from the original groundtruth of M classes.

to construct the graph and perform extensive experiments to verify its validity. Intuitively, unlike standard convolutions which operates on a local regular grid, the graph enables us to compute the response of a node based on its neighbors defined in the adjacency matrix, hence receiving a much wider receptive field than regular convolutions. Formally, the graph convolution is defined as,

$$Z = \sigma(AXW), \tag{1}$$

where $\sigma(\cdot)$ denotes the non-linear activation function, $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix measuring the relations of nodes in the graph and $W \in \mathbb{R}^{D\times D}$ is the weight matrix. In our experiments, we use ReLU as activation function and perform experiments with different graph construction methods.

B. Overall Framework

As illustrated in Fig. 2, we present the Boundary-aware Graph Convolution Network to harvest and propagate features within the object regions isolated by the learned boundaries. We use the ResNet-101 pretrained on the ImageNet dataset as the backbone, replace the last two down-sampling operations and employ dilation convolutions in the subsequent convolutional layers, hence enlarging the resolution and receptive field of the feature map, so the output stride becomes 8 instead of 16, which preserves more details.

Our model consists of two parts: basic network and BGC module. Specifically, we adopt ResNet-101 together with atrous spatial pyramid pooling(ASPP) as the basic complete segmentation network. An input image is passed through the backbone and ASPP module, then produces a feature map $X \in \mathbb{R}^{C\times H\times W}$, where C, H, W represent channel number, height and width respectively. Then we apply a convolution layer to realize the dimension reduction and the feature X will participate in two different branches, The first branch is the prediction step which produces the binary boundary prediction map. The binary boundary map and the feature X are subsequently fed into the BGC module to perform boundary-aware graph convolution. And the output feature of our BGC module is concatenated with the input feature, and refine it through a 1×1 convolution to get the final refined segmentation result.

C. Boundary Learning

Lots of works have contributed to boundary detection problem [24]–[27], but most of them focus on edges that sketch the objects. Different from them, we only focus on the boundaries of semantic objects predefined in the dataset. The boundary label can be easily generated from the existing groundtruth, as shown in Fig.3. Specifically, we treat the boundary as an additional semantic category and learn it simultaneously with other categories. As shown in Fig.2, we obtain a new groundtruth(M+1 classes) from the original one(M classes) and utilize it for supervising the network(Loss 2) and infer the boundary layout.

Different from previous works that treat boundary detection and semantic segmentation as two separate tasks, our proposed boundary learning method is highly embedded with semantic segmentation. Moreover, our boundary detection method only targets at boundaries of semantic objects predefined in the datasets instead of all the edges in the image where dramatic changes in color or texture take place. Since what we focus on is the boundaries that segment different objects or regions in the image so that feature similarity can be enhanced within an object. More importantly, these two tasks are combined into one branch, they can benefit each other. Concisely, by training them together, the semantic segmentation information can help suppress the edges in the object which do not belong to the semantic boundary of the object, while boundary information can assist in semantic segmentation with more attention to boundary layout.

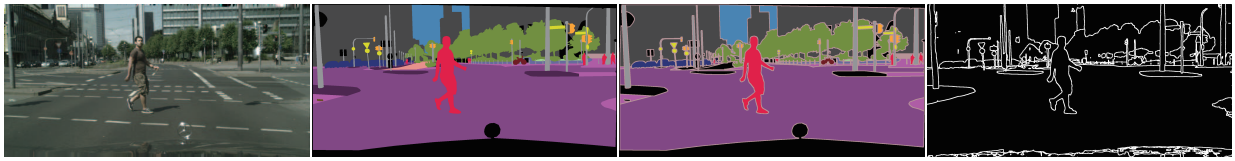

Fig. 3. From left to right: input image, original ground truth(M classes), generated ground truth(M+1 classes), learned boundary map

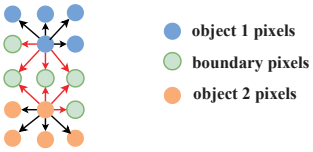

Fig. 4. Illustration of boundary-aware sampling method. Black line denotes the normal connection and red line denotes the canceled connection.

D. Boundary-aware Graph Convolution

The detailed structure of BGC module is shown in Fig.2. It consists of two subsequent processes, including graph construction and graph reasoning. The proposed module is based on a coarse-to-fine framework, where the input is the feature map X, boundary prediction map and the output is the reasoned feature map to be concatenated with the original feature.

1) Graph Construction: **Similarity Graph.** Intuitively, we can build the graph(which is adjacency matrix in our formulation) based on the similarity(edge weight) between different nodes, for two node features x_i, x_j, the pairwise similarity between this two nodes is defined as,

$$F(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i})^T \phi'(\boldsymbol{x_j}), \tag{2}$$

where ϕ, ϕ'denote two different transformations of the original features. In practice, we adopt linear transformations, hence $\phi(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x}$ and $\phi'(\boldsymbol{x}) = \boldsymbol{w'}\boldsymbol{x}$. The parameters $\boldsymbol{w}$ and $\boldsymbol{w'}$ are both $D \times D$ dimensions weights which can be learned via back propagation, forming a dynamically learned graph construction method. After computing the similarity matrix, we perform normalization on each row of the matrix so that the sum of all the edge values connected to one node i will be 1. In practice, we choose softmax as normalization function,so the output adjacency matrix will be,

$$A_{ij} = \frac{exp(F(\boldsymbol{x_i}, \boldsymbol{x_j}))}{\sum_{j=1}^{N} exp(F(\boldsymbol{x_i}, \boldsymbol{x_j}))} \tag{3}$$

Boundary-aware sampling. We can first build a neighbors-connected graph(adjacency matrix) with the formulation introduced above with similarity value as the edge weight, where a node in the feature map is only connected with its eight geographical neighbors. In order to obstruct the information flow across the boundaries, we utilize the predicted binary boundary map obtained by the method discussed above where 0/1 denotes the existence of the boundary. Specifically, for every node in the graph, if it belongs to the boundary, the edges connecting it with other nodes are canceled whose weights equal to 0 in practice. In this way, for one object, the edges connecting its inside pixels with outside pixels get impeded, forming a boundary-aware feature propagation manner, where information flow only takes place inside an object, as shown in Fig.4. [33] proposes DAG-RNN to capture long-range context based on directed acyclic graphs(DAGs), where the directed connections of each pixel is specified, which is similar to our proposed feature propagation method. However, DAG-RNN has to scan the image pixel by pixel and requires a lot of loops and is much difficult to implement, hence not practical in real-world applications, while our proposed boundary-aware sampling achieves this objective in a very simple way. With a simple graph construction process, information flow can be controlled as designed, hence enhancing the feature similarity within the object and keeping discrimination from different objects.

Graph Reasoning. Discriminative pixel-level feature representations are essential for semantic segmentation, which could be obtained by the proposed graph convolution based module in a boundary-aware manner. By exploiting the relations between pixels sampled by the above method, the inside-object consistency can be preserved and moreover, outside-object discrepancy can also be enhanced. As shown in Fig.2, the module takes the coarse feature map $X \in \mathbb{R}^{C \times H \times W}$ as input, where C, H, W denote the dimension of the feature map, height and width respectively. Inspired by point cloud segmentation [34], [35] ,we treat nodes in the feature map as the vertexes in the graph. Therefore, we transform the feature map to the graph representation $X \in \mathbb{R}^{C \times N}$, where $N = H \times W$ denotes the number of nodes in the feature map. Applying the graph construction method discussed above, we can obtain the adjacency matrix $A \in \mathbb{R}^{N \times N}$ of the feature map. Following the paradigm of graph convolution, we get the reasoned feature $X \in \mathbb{R}^{C \times N}$ and then reshape it back to the original grid form $X \in \mathbb{R}^{C \times H \times W}$. Once obtaining the reshaped reasoned feature, we combine this feature map with the input feature map to get the final output(refined feature). Specifically, the combine method is concatenation. Since the reasoned feature mainly targets at enhancing feature similarity while the original original feature still preserves discrimination. Finally, the refined feature is passed through the conventional 1×1 convolution layer to get the final segmentation prediction map.

E. Loss Function

Both coarse and refined output are supervised with the semantic labels. Furthermore, following normal practice in previous state-of-the-art works [3], [36], [37], we add the auxiliary supervision for improving the performance, as well as making the network easier to optimize. Specifically, the output of the third stage of our backbone ResNet-101 is further fed into a auxiliary layer to produce a auxiliary prediction, which is supervised with the auxiliary loss. As for the main path, coarse segmentation result and refined segmentation result are produced and hence require proper supervision. Concisely, the numbers of categories of the coarse prediction, refined prediction and auxiliary prediction are $M+1, M, M$ respectively, where M is the number of categories of the dataset. We apply standard cross entropy loss to supervise the auxiliary output and the coarse prediction map, and employ OHEM loss [38] for the refined prediction map. In a word, the loss can be formulated as follows,

$$L = \alpha \cdot l_c + \beta \cdot l_f + \gamma \cdot l_a \tag{4}$$

where α, β, γ are used to balance the coarse prediction loss l_c, refined prediction loss l_f and auxiliary loss l_a.

IV. Experiments

To evaluate the performance of our proposed BGC module, we carry out extensive experiments on benchmark datasets including Cityscapes [6], PASCAL VOC 2012 [7] and COCO Stuff [8]. Experimental results demonstrate that the proposed method can effectively boost the performance of the state-of-the-art methods. In the following section, we will introduce the datasets and implementation details, and then perform ablation study on Cityscapes dataset. Finally, we report the results on PASCAL VOC 2012 dataset and COCO Stuff dataset.

A. Datasets and Evaluation Metrics

1) Cityscapes: The Cityscapes dataset [6] is tasked for urban scene understanding, which contains 30 classes and only 19 classes of them are used for scene parsing evaluation. The dataset contains 5000 finely annotated images and 20000 coarsely annotated images. The finely annotated 5000 images are divided into 2975/500/1525 images for training, validation and testing.

2) PASCAL VOC 2012: The PASCAL VOC 2012 dataset [7] is one of the most competitive semantic segmentation dataset which contains 20 foreground object classes and 1 background class. The dataset is split into 1464/1449/1556 images for training, validation and testing. [39] has augmented this dataset with annotations, resulting in 10582 train-aug images.

3) COCO Stuff: The COCO Stuff dataset [8] is a challenging scene parsing dataset containing 59 semantic classes and 1 background class. The training and test set consist of 9K and 1K images respectively.

In our experiments, the mean of class-wise Intersection over Union (mIoU) is used as the evaluation metric.

Method	mIOU(%)
ResNet-101 Baseline	76.3
ResNet-101 + plain GCN	78.2
ResNet-101 + boundary-aware GCN	79.9

TABLE I
PERFORMANCE COMPARISONS OF OUR PROPOSED BOUNDARY-AWARE GCN AND PLAIN GCN ON CITYSCAPES VALIDATION SET.

B. Implementation Details

We choose the ImageNet pretrained ResNet-101 as our backbone and remove the last two down-sampling operations, and employ dilated convolutions in the subsequent convolution layers, making the output stride equal to 8. For training, we use the stochastic gradient descent(SGD) optimizer with initial learning rate 0.01, weight decay 0.0005 and momentum 0.9 for Cityscapes dataset. Moreover, we adopt the 'poly' learning rate policy, where the initial learning rate is multiplied by $(1-\frac{iter}{max\_iter})^{power}$ with power=0.9. For Cityscapes dataset, we adopt the crop size as 769×769, batch size as 8 and training iterations as 30K. For PASCAL VOC 2012 dataset, we set the initial learning rate as 0.001, weight decay as 0.0001, crop size as 513×513, batch size as 16 and training iterations as 30K. For COCO Stuff dataset, we set initial learning rate as 0.001, weight decay as 0.0001, crop size as 520×520, batch size as 16, and training iterations as 60K.

C. Ablation Study

In this subsection, we conduct extensive ablation experiments on the validation set of Cityscapes dataset with different settings for our proposed BGCNet.

1) The impact of boundary-aware sampling: We use the dilated ResNet-101 as the baseline network, and final segmentation result is obtained by directly upsampling the output. To evaluate the effectiveness of the proposed boundary-aware strategy, we carry out the experiments where plain GCN and boundary-aware GCN are adopted separately. Concretely, plain GCN is realized by simply performing graph convolution operation on the feature map obtained from the backbone, where fully-connected manner is implemented. While boundary-aware GCN is realized in a boundary-aware manner, where geographical neighbors are connected apart from boundary pixels. As shown in Table I, the proposed boundary-aware GCN reasoning performs better than the plain GCN. Since plain GCN adopts fully connected fashion onto the input feature map, it serves similarly as self-attention mechanism, where boundary details are ignored while our method can effectively overcome this issue.

2) The impact of BGC module: Based on the dilated ResNet-101 backbone, we subsequently add ASPP module and the proposed module to evaluate the performance, as shown in Table II. The result of solely adding ASPP module is 78.4%, which is 1.5% lower than solely adding BGC module. Finally, we choose ResNet-101 plus ASPP module as our basic segmentation network and use BGC module

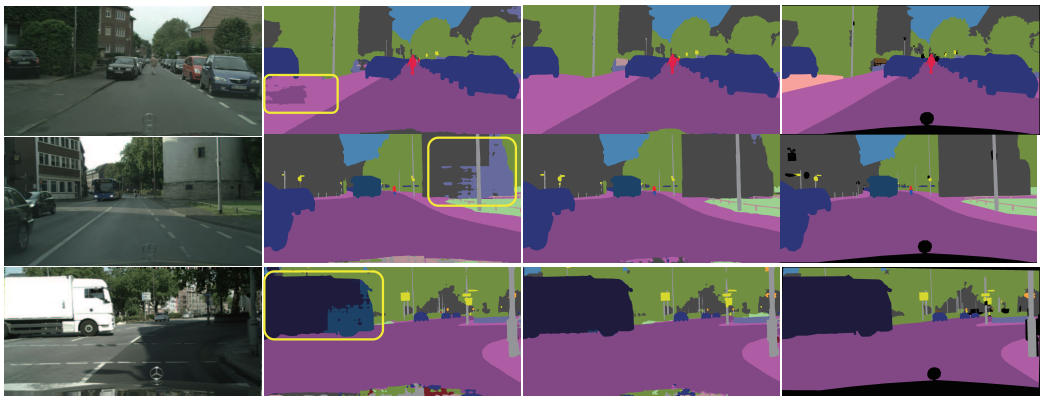

Fig. 5. Visualization results on Cityscapes val set, where yellow squares denote the challenging regions that can be resolved by our proposed BGC module.

Method	mIOU(%)
ResNet-101 Baseline	76.3
ResNet-101 + ASPP	78.4
ResNet-101 + BGC	79.9
ResNet-101 + ASPP + BGC	81.1

TABLE II
DETAILED PERFORMANCE COMPARISONS OF OUR PROPOSED BOUNDARY-AWARE GRAPH CONVOLUTION MODULE ON CITYSCAPES VALIDATION SET.

to further refine the result, achieving 2.7% gain in mIOU over the basic network, which demonstrates that our proposed BGC module can be easily plugged into any state-of-art segmentation network to further boost the performance. The effect of BGC module can be visualized in Fig.5. Some details and boundaries are refined compared to the coarse prediction map output by the basic network. Results demonstrate that, with the boundary information, the BGC module build stronger connections of pixels within the same segment, thus features of the same segment become more similar while features of different segments remain discriminative.

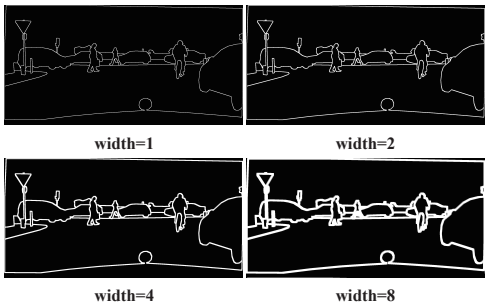

Fig. 6. Trimaps used for boundary accuracy evaluation with different band width from Cityscapes validation set.

3) The impact on boundary accuracy: To validate the effectiveness of BGC module, we design an experiment to show that our method improves boundary precision. Following [40], [41], we adopt a standard trimap approach where we compute

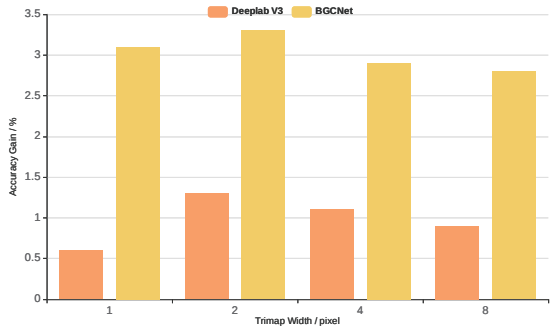

Fig. 7. Boundary accuracy gain of deeplab v3 and BGCNet over baseline

Method	MS	Flip	mIOU(%)
BGCNet			81.1
BGCNet	✓		81.6
BGCNet		✓	81.4
BGCNet	✓	✓	**81.9**

TABLE III
PERFORMANCE INFLUENCES WITH DIFFERENT EVALUATION STRATEGIES ON CITYSCAPES VALIDATION SET.

the classification accuracy within a band (called trimap) of varying width around boundaries of segments(shown in Figure 6). As shown in Figure 7, we use dilated ResNet-101 as baseline and compute the boundary accuracy gain over baseline of deeplab v3 and BGC module. The results indicate that BGC module effectively improves the vicinity of boundaries compared with baseline, hence preserving more details.

4) The impact of evaluation strategies: Based on details discussed above, we propose Boundary-aware Graph Convolution Network (BGCNet) with ResNet-101+ASPP as basic network. Like previous work [3], [42]–[44], we also adopt the left-right flipping and multi-scale [0.75, 1.0, 1.25, 1.5, 1.75, 2.0] evaluation strategies. From Table III, MS/Flip improves

Method	Backbone	mIOU(%)
DeepLab-v2 [16]	ResNet-101	70.4
RefineNet [14]	ResNet-101	73.6
GCN [45]	ResNet-101	76.9
SAC [46]	ResNet-101	78.1
PSPNet [3]	ResNet-101	78.4
BiSeNet [47]	ResNet-101	78.9
AAF [48]	ResNet-101	79.1
DFN [49]	ResNet-101	79.3
PSANet [4]	ResNet-101	80.1
DenseASPP [42]	DenseNet-161	80.6
GloRe [22]	ResNet-101	80.9
DANet [43]	ResNet-101	81.5
BGCNet(Ours)	ResNet-101	**82.1**

TABLE IV
COMPARISONS WITH STATE-OF-ART ON THE CITYSCAPES TEST SET.

Method	Backbone	mIOU(%)
FCN [1]	VGG-16	62.2
DeepLab-CRF [16]	VGG-16	71.6
PSPNet [3]	ResNet-101	82.6
DFN [49]	ResNet-101	82.7
DANet [43]	ResNet-101	82.6
EncNet [50]	ResNet-101	82.9
BGCNet(Ours)	ResNet-101	**84.2**

TABLE V
COMPARISONS WITH STATE-OF-ART ON THE PASCAL VOC 2012 TEST SET.

the performance by 0.8% on validation set.

5) Comparisons with state-of-the-arts: Furthermore, we also train the proposed BGCNet using both training and validation set and make the evaluation on the test set by submitting our test results to the official evaluation server. Specifically, we use dilated ResNet-101 as backbone. Moreover, we use the multi-scale and flip strategies while testing. From Table IV, it can be observed that our BGCNet achieves state-of-the-art performance on Cityscapes test set.

D. Experiments on PASCAL VOC 2012

We carry out experiments on the PASCAL VOC 2012 dataset to further evaluate the effectiveness of our method. We use train-aug and validation set for better training. Quantitative results of PASCAL VOC 2012 test set are shown in Table V. It can be seen that our method achieves state-of-the-art result on the test set without COCO dataset pretraining.

E. Experiments on COCO Stuff

We also conduct experiments on the COCO Stuff dataset and report the related results in Table VI. Results show that our model achieves 40.7% in mean IOU which is the highest record. Hence our method can effectively collect useful long-range contextual information and obtain better feature representation in semantic segmentation.

V. CONCLUSION

In this paper, in order to enhance feature similarity within the object while keeping feature discrimination of different objects, we present the Boundary-aware Graph Convolution Network(BGCNet). Specifically, we first propose a boundary learning method which learns boundary information simultaneously with semantic segmentation task, where two tasks can benefit each other during learning process. Then the boundary-aware graph convolution(BGC) module utilizes boundary prediction for graph construction, where information flow takes place inside the segments. After graph convolution, the reasoned feature is fused with the coarse feature to obtain the final refined feature, subsequently producing the final refined segmentation result. The ablation experiments demonstrate the effectiveness of BGC module. Our BGCNet achieves outstanding performance on three benchmark datasets, *i.e.*Cityscapes, PASCAL VOC 2012 and COCO Stuff.

Method	Backbone	mIOU(%)
FCN-8s [1]	VGG-16	22.7
DAG-RNN [33]	VGG-16	31.2
RefineNet [14]	ResNet-101	33.6
CCL [51]	ResNet-101	35.7
DSSPN [52]	ResNet-101	37.3
SGR [23]	ResNet-101	39.1
BGCNet(Ours)	ResNet-101	**41.7**

TABLE VI
COMPARISONS WITH STATE-OF-ART ON THE COCO STUFF TEST SET.

ACKNOWLEDGMENT

This work was supported by the National Key R&D Program of China under grant 2017YFB1002804 and National Natural Science Foudation of China (No.31771230).

REFERENCES

[1] 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*, 2015, pp. 3431–3440.
[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
[3] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
[4] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 267–283.
[5] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.
[6] 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*, 2016, pp. 3213–3223.
[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
[8] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1209–1218.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] 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*, 2016, pp. 770–778.

[12] 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*. Springer, 2015, pp. 234–241.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[14] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[15] 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.

[16] ——, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[17] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

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

[19] S. Chandra, N. Usunier, and I. Kokkinos, "Dense and low-rank gaussian crfs using deep embeddings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5103–5112.

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

[21] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 399–417.

[22] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 433–442.

[23] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 1853–1863.

[24] G. Bertasius, J. Shi, and L. Torresani, "High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 504–512.

[25] W. Shen, B. Wang, Y. Jiang, Y. Wang, and A. Yuille, "Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2391–2400.

[26] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3982–3991.

[27] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[28] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4545–4554.

[29] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5696–5704.

[30] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5229–5238.

[31] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary loss for highly unbalanced segmentation," *arXiv preprint arXiv:1812.07032*, 2018.

[32] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," *arXiv preprint arXiv:1912.08193*, 2019.

[33] B. Shuai, Z. Zuo, B. Wang, and G. Wang, "Scene segmentation with dag-recurrent neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1480–1493, 2017.

[34] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[35] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, p. 146, 2019.

[36] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 593–602.

[37] F. Zhang, Y. Chen, Z. Li, Z. Hong, J. Liu, F. Ma, J. Han, and E. Ding, "Acfnet: Attentional class feature network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6798–6807.

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

[39] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 991–998.

[40] P. Kohli, P. H. Torr *et al.*, "Robust higher order potentials for enforcing label consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[41] D. Marin, Z. He, P. Vajda, P. Chatterjee, S. Tsai, F. Yang, and Y. Boykov, "Efficient segmentation: Learning downsampling near semantic boundaries," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2131–2141.

[42] 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*, 2018, pp. 3684–3692.

[43] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[44] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," *arXiv preprint arXiv:1811.11721*, 2018.

[45] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[46] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2031–2039.

[47] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 325–341.

[48] T.-W. Ke, J.-J. Hwang, Z. Liu, and S. X. Yu, "Adaptive affinity fields for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 587–602.

[49] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1857–1866.

[50] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[51] H. Ding, X. Jiang, B. Shuai, A. Qun Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2393–2402.

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