2021 20th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Multiscale Global Channel Network for Edge Detection

1st Wang Zhang
School of Artifical Intelligence and Computer Science
Jiangnan University
Wuxi, China
forzhangwang @163.com

2nd Wei Fang
School of Artifical Intelligence and Computer Science
Jiangnan University
Wuxi, China
fangwei@jiangnan.edu.cn

3rd Jun Sun
School of Artifical Intelligence and Computer Science
Jiangnan University
Wuxi, China
junsun@jiangnan.edu.cn

4th Qidong Chen
School of Artifical Intelligence and Computer Science
Jiangnan University
Wuxi, China
cqd_mu@hotmail.com

***Abstract*—Neural network is used to fuse the spatial and channel information of each layer's local receptive fields to construct information features. But global long-range dependency is not effectively modeled, which leads to non-optimal discriminative feature representations. In this paper, we propose multi-scale global channel network (MSGC). We use self-attention mechanism to combine local features with their corresponding global dependencies, adaptively recalibrate the channel response, guide the network to ignore irrelevant information, and emphasize the correlation of relevant features. We evaluated the proposed method on BSDS500 dataset and NYUD dataset. MSGC achieves ODS F-measure of 0.815 on BSDS500, which is 0.9% higher than the existing technology.**

Keywords-Convolutional Neural Networks; Deep learning; Edge detection; Deep attention; Self-attention

I. INTRODUCTION

Edge detection aims to extract perceptually salient edges of natural images, which is important to high level computer vision tasks, such as image segmentation [1], [2], object detection/recognition[3], [4].

The early traditional methods include Sobel detector [5], widely used Canny detector [6],Structured Edges [7] and gPb [2]. CNN is used for edge detection, including DeepContour [8] and CSCNN [9]. HED [10] and RCF [11] supervise the predictions of different network layers. Richer convolution features are very effective for many visual tasks, but HED and RCF still do not explicitly use global context information, and do not directly impose constraints on adjacent pixel labels to enhance depth supervision. Therefore, we can improve the quality of network representation by explicitly modeling the dependency of channels.

Because convolution layer establishes the pixel relationship in the local neighborhood[12],[13], the modeling of long-range dependency is invalid. We add features of all positions to the features of each location.

II. RELATED WORK

This paper mainly studies edge detection and deep attention. We briefly review the related work in these two aspects.

A. Edge Detection

Edge detection is one of the most basic and challenging problems in computer vision.

These methods can be roughly divided into three categories: traditional edge operators, learning based methods and deep learning based methods. The traditional edge operator detects edges by detecting abrupt changes in intensity, color and texture. Sobel [5] applied thresholding the gradient of the image. The learning based method uses hand-crafted features. Arbeláez *et al.* [1] combined local clues into a global framework. In recent years, advanced results have been obtained by using deep learning to extract depth features automatically. Xie and Tu [10] proposed an end-to-end model for in-depth monitoring different scale features of side outputs. Liu *et al.* [11] connected the side outputs to all the convolution layers of VGG16 [14]. MSGC is based on RCF [11]. The above training strategy does not explicitly use context information and impose constraints on adjacent pixel labels. We use global features to enhance context modeling for multiscale side outputs.

B. Deep Attention

The attention mechanism aims to emphasize the important areas and filter irrelevant information. It has been successfully applied to various visual tasks, such as classification [15] and detection [16]. PSANET [17] adaptively linked each location in the feature map with other locations. Senet [13] and Genet [18] recalibrated channel dependencies by rescaling different channels. However, the feature fusion method is not effective enough. In this paper, addition fusion is used to model the global context more effectively.

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00033

III. METHODS

A. Overview

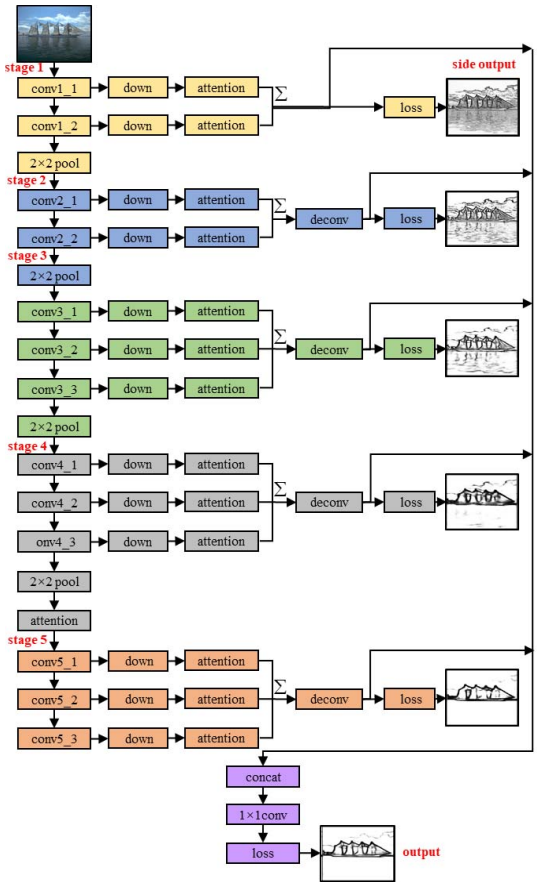

Figure 1. The architecture of MSGC.

VGG16 network is composed of 13 convolution layers, 3 fully connected layers and 5 downsampling layers. We make following changes to VGG16: (1) because the fifth pooling layer produces a too fuzzy prediction map to be used, the fifth pooling layer and all fully connected layers are discarded, (2) a downsampling layer is connected to each convolution layer to extract different scale features, (3) fuse the multiscale features of each stage, then the deconvolution layer up-sample the fused features, (4) a convolution layer is used to fuse the side outputs.

Let (X,Y) denotes one sample of input training data set T, where $X=\{x_i, i=1,\cdots,|X|\}$ is a raw input image and $\mathrm{Y}=\{\mathrm{y}_i,\mathrm{i}=1,\cdots,|\mathrm{X}|\}$, $\mathrm{y}_j\in\{0,1\}$ is the corresponding ground truth edge map. The training loss for every image is formulated as

$$l(X;W)=\alpha\sum_{i\in Y_+}\log P(y_i=1\mid X;W)+\beta\sum_{i\in Y_-}\log(1-P(y_i=0\mid X;W)), \quad (1)$$

where $\alpha=\lambda\cdot\frac{|\mathrm{Y}_-|}{|\mathrm{Y}_+|+|\mathrm{Y}_-|}$, $\beta=\frac{|\mathrm{Y}_+|}{|\mathrm{Y}_+|+|\mathrm{Y}_-|}$, Y_+ and Y_- denote the edge and non-edge ground truth label sets respectively, λ is to automatically balance the loss between positive/negative classes, and W denotes all the network layers parameters. The final loss can be obtained by further aggregating these generated edge maps, *i.e.*,

$$\mathrm{L}(\mathrm{W})=\sum_{j=1}^{5}l(X^j;W)+l(X^{fuse};W), \quad (2)$$

where X^{j} denotes the edge map of stage j and X^{fuse} denotes the edge map of fusion layer.

Traditional CNNs has a local receptive field, so the generated local features may cause potential differences between features of pixels with the same label. We study the self-attention mechanism of establishing association between features. First, the global context information is captured. Then, the learned global features are input into the channel self-attention module. The self-attention module helps to adaptively combine local features with their global context, and can gradually filter out noise by emphasizing useful information. The overview of the proposed architecture is depicted in Figure 1.

B. Global Channel Self-attention Modules (GCM)

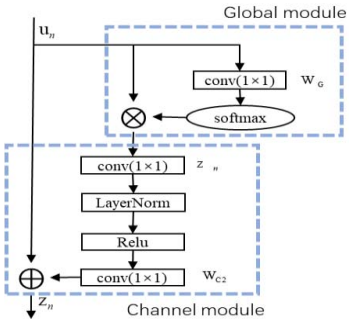

Figure 2. Architecture of the GCM.

Firstly, we use 1×1 convolution $\mathrm{W_G}$ and softmax function to obtain attention weights, and compute a global context attention map S. Then we recalibrate the channel response through 1×1 convolution $\mathrm{W_c}$. Finally, we aggregate the global context features to the features of each location by addition. We use $\mathrm{U}=\{\mathrm{u}_n,\mathrm{n}=1,\cdots,\mathrm{N}\}$ as an input feature map, where $\mathrm{N}=\mathrm{H}\times\mathrm{W}$ is the number of positions in the feature map. Our global attention map is formulated as follows:

$$\mathrm{S}=\sum_{n=1}^{N}\frac{\mathrm{f}(u_n)}{C(U)}u_n, \quad (3)$$

where n lists all possible locations, $\mathrm{f}(u_n)=\mathrm{e}^{W_g\mathrm{u}_n}$ is an embedded Gaussian function to calculate the similarity in the embedding space, $C(U)=\sum_{\mathrm{m}=1}^{N}\mathrm{e}^{W_g\mathrm{u}_m}$ is a normalization factor.

In order to make the GCM lightweight, we use the bottleneck transform module. We add layer normalization in bottleneck transformation before ReLU layer to simplify optimization, and also play the role of regulation, which is illustrated in Figure 2. We denote $Z=\{z_n, n=1,\cdots, N\}$ as the output feature maps of our attention module, the complete GCM can be expressed as:

$$z_n = u_n + W_{C2}\,\mathrm{Re}lu(LN(W_{C1}S)). \quad (4)$$

IV. EXPERIMENTS

A. Datasets

BSDS500 contains 200 training, 100 verification and 200 test images. We expanded the training set and verification set with rotation, flipping, scaling. We mix the enhanced data of with the flipped Pascal VOC context dataset [19] as training data with 49006 training.

The NYUD [20] dataset consists of 1449 pairs of aligned RGB and depth images. We only use the RGB part. We split the NYUD dataset into 381 training, 414 validation, and 654 test images[21], and expand them by randomly flipping, scaling and rotating.

B. Implementation Details

We implement MSGC using PyTorch. The VGG16 pretrained on ImageNet [22] is used to initialize MSGC. The threshold λ used for loss computation is set as 1.1 and 1.2 for BSDS500 and NYUD dataset, respectively.

SGD optimizer randomly extracts 10 images in each iteration, and the global learning rate is set to 1e-6, which decreases 10 times after every 10K iterations. Momentum and weight decay are set to 0.9 and 0.0002, respectively. We do a total of 40K iterations. All experiments in this paper are performed with NVIDIA 1080 GPU.

Before evaluation, we used non-maximum suppression (NMS) to refine the edge. The maximum allowed tolerance between edge predictions and ground truth for BSDS500 and NYUD dataset are set to 0.0075 and 0.011 respectively.

C. Comparison with Other Works

Performance on BSDS500： Our experimental results with several state-of-the-art edge detection networks on BSDS500 are summarized in 0 and Figure 3.

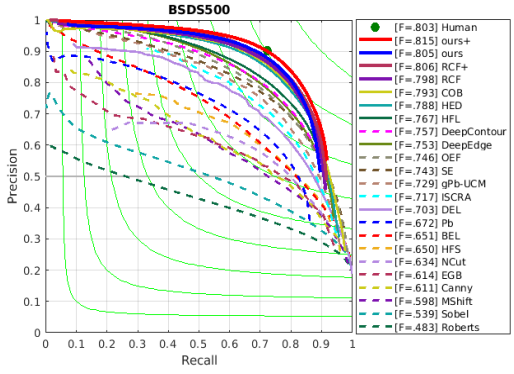

Figure 3. The p-r curves of MSGC and other works on BSDS500 dataset.

TABLE I. The comparison with other methods on BSDS500 dataset. +indicates trained with additional PASCAL VOC Context dataset.

Methods	ODS	OIS	AP
Human	0.803	0.803	-
Canny [6]	0.611	0.676	0.520
SE [7]	0.743	0.763	0.800
OEF [23]	0.746	0.770	0.820
DeepEdge [6]	0.753	0.769	0.784
DeepContour [27]	0.757	0.776	0.790
HFL [29]	0.767	0.788	0.795
HED [10]	0.788	0.808	0.840
CEDN+ [25]	0.788	0.804	-
RDS [29]	0.792	0.810	0.818
RCF [11]	0.798	0.815	-
RCF [11]+	0.806	0.824	0.840
DeepBoundary [27]	0.789	0.811	0.789
DeepBoundary+ [27]	0.809	0.827	0.861
MSGC	**0.805**	**0.822**	0.834
MSGC+	**0.815**	**0.834**	**0.866**

As shown in the results, GCM can actually improve the performance of edge detection. Figure 4. shows a comparison of edge maps from MSGC and RCF before NMS. MSGC can effectively eliminate most of the blurred and noisy boundaries and produce clearer image edges.

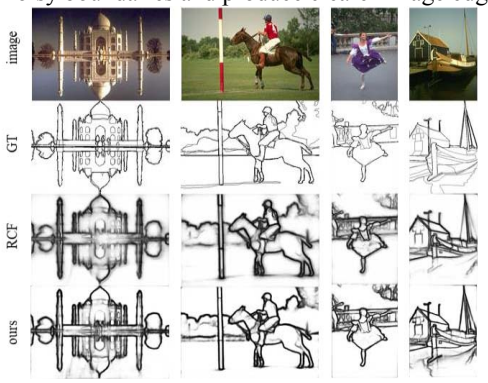

Figure 4. Comparison of edge maps before NMS on BSDS500 dataset.

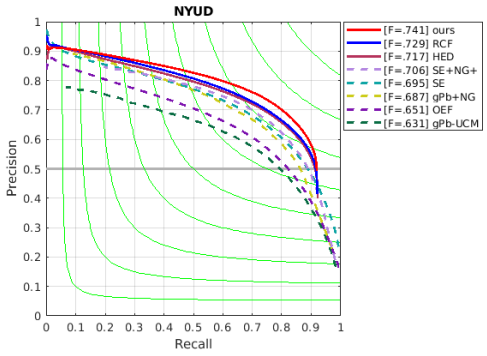

Figure 5. The p-r curves of MSGC and other works on NYUD dataset

TABLE II. Comparison with other methods on NYUD dataset.

Methods	ODS	OIS	AP
gPb-UCM [1]	0.631	0.661	0.562
gPb+NG [21]	0.687	0.716	0.629
OEF [23]	0.651	0.667	-
SE [7]	0.695	0.708	0.679
SE+NG+ [28]	0.706	0.734	0.738
HED [10]	0.717	0.732	0.734
RCF [11]	0.729	0.742	-
LPCB [29]	0.739	0.754	-
MSGC	**0.741**	**0.759**	**0.740**

Performance on NYUD： TABLE II. and Figure 5. show the quantitative results of MSGC compared with several recent methods. MSGC achieves the best performance of ODS F-score 0.741, which proves the effectiveness of MSGC.

V. CONCLUSION

In this paper, we introduce a deep attention architecture to complete the edge detection task. It combines different levels of global information with GCM to model long-range dependency effectively. Finally, a dynamic channel-feature recalibration is performed to filter the noisy regions and help the network focus on the relevant areas in the image. MSGC is compared with more than 10 edge detection methods on BSDS500 dataset and NYUD dataset, and MSGC provides accurate and reliable edge detection.

ACKNOWLEDGMENT

We wish to thank every member of the team for their efforts.

REFERENCES

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. IEEE TPAMI,33(5):898–916, 2011.

[2] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interac-tive foreground extraction using iterated graph cuts. In IEEE Trans. Graph., volume 23, pages 309–314. ACM, 2004.

[3] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, pages 91–99, 2015.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, volume 1, pages 886–893. IEEE, 2005.

[5] J. Kittler. On the accuracy of the sobel edge detector. Image and Vision Computing, 1(1):37–42, 1983.

[6] J. Canny. A computational approach to edge detection. IEEE TPAMI, 8(6):679–698, 1986.

[7] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. IEEE TPAMI, 37(8):1558–1570, 2015.

[8] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. DeepContour: A deep convolutional feature learned by positivesharing loss for contour detection. In IEEE CVPR, pages 3982–3991, 2015.

[9] J.-J.Hwang and T.-L.Liu. Pixel-wise deep learning for contour detection. arXiv preprint arXiv:1504.01989, 2015.

[10] S. Xie and Z. Tu. Holistically-nested edge detection. In IJCV. Springer, 2017.

[11] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, Jinhui Tang. Richer Convolutional Features for Edge Detection. IEEE TPAMI, 2019.

[12] X.Wang,R.Girshick,A.Gupta,andK.He. Non-local neural networks. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018.

[13] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In IEEE Conference on Computer Vision and Pattern Recognition, 2018.

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

[15] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.

[16] H. Li, Y. Liu, W. Ouyang, and X. Wang, "Zoom out-and-in network with map attention decision for region proposal and object detection," International Journal of Computer Vision, vol. 127, no. 3, pp. 225–238, 2019.

[17] 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.

[18] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In Advances in Neural Information Processing Systems, pages 9423–9433, 2018.

[19] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Y uille. The role of context for object detection and semantic segmentation in the wild. In IEEE CVPR, pages 891–898, 2014.

[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In ECCV, 2012.

[21] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In CVPR, 2013.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: Alarge-scalehierarchicalimagedatabase. In IEEE CVPR, pages 248–255. IEEE, 2009.

[23] S. Hallman and C. C. Fowlkes. Oriented edge forests for boundary detection. In IEEE CVPR, pages 1732–1740, 2015.

[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 IEEE ICCV, pages 504–512, 2015.

[25] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang. Object contour detection with a fully convolutional encoder-decoder network. arXiv preprint arXiv:1603.04530, 2016.

[26] Y. Liu and M. S. Lew. Learning relaxed deep supervision for better edge detection. In IEEE CVPR, pages 231–240, 2016.

[27] Y. Wang, X. Zhao, and K. Huang. Deep crisp boundaries. In CVPR, 2017.

[28] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-dimages for object detection and segmentation. In ECCV, pages 345–360. Springer, 2014.

[29] R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu. Learning to predict crisp boundaries. In ECCV, pages 562–578, 2018.