CAN: CONTEXTUAL AGGREGATING NETWORK FOR SEMANTIC SEGMENTATION

Dechun Cong[1,2], *Quan Zhou*[1,2,*], *Jie Cheng*[3], *Xiaofu Wu*[1], *Suofei Zhang*[4], *Weihua Ou*[5], *and Huimin Lu*[6]

[1]National Engineering Research Center of Communications and Networking,
Nanjing University of Posts & Telecommunications, P.R. China.
[2]State Key Lab. for Novel Software Technology, Nanjing University, P.R. China.
[3]Huawei Technologies Co. Ltd., P.R. China.
[4]School of Internet of Things, Nanjing University of Posts & Telecommunications, P.R. China.
[5]School of Big Data and Computer Science, Guizhou Normal University, P.R. China.
[6]Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Japan.

ABSTRACT

Fully convolutional neural networks (FCNs) have shown great success in dense estimation tasks. One key pillar of such progress is mining multi-scale context cues from features in different convolutional layers. This paper introduces *contextual aggregating network(CAN)*, a generic convolutional feature ensembling framework for semantic segmentation. Our framework first captures multi-scale contextual clues by concatenating multi-level feature representation, which carries both coarse semantics and fine details. Then it adaptively integrates stacked features to perform dense pixel estimation. The proposed CAN is trainable end-to-end, and allows us to *fully* investigate multi-scale context information embedded in images. The experiments show the promising results of our method on PASCAL VOC 2012 and Cityscapes dataset.

Index Terms— Semantic segmentation, Convolutional features, Fully convolutional networks, Multi-scale context

1. INTRODUCTION

Semantic segmentation plays an important role in image understanding. The task here is to assign a category label for each image pixel, which thus can be also considered as a dense prediction problem. There are two sub-tasks for image semantic segmentation: (1) classification, where a unique semantic concept should be marked correctly to the associated object; (2) localization, where the assigned label for pixel must be aligned to the appropriate coordinates in the segmentation output. To this end, a well-designed segmentation system should simultaneously deal with these two issues.

Recently, due to the powerful ability to abstract high-level semantics from raw images, deep learning based approaches, especially the convolutional neural networks (CNNs), e.g., VGG-based fully convolutional neural networks (FCNs) [1, 2, 3], residual networks (RENs) [4, 5, 6], and deconvolutional networks (DENs) [7, 8, 9], have achieved remarkable progress for the task of semantic segmentation. However, these methods have some shortcomings when they deal with dense labeling tasks. In FCNs and RENs, multiple stages of spatial pooling and convolution stride significantly reduce the dimension of feature representation, thereby losing much of the finer image structure. This invariance to local image transformation is helpful for image classification [10, 11], but may be harmful for the task of semantic segmentation [2, 3]. In order to address this problem, DEN-based networks have been proposed in recent years, where the up-sampling operation is employed to produce high-resolution feature maps by learning deconvolutional filters [7, 8]. These approaches, however, still suffer from a couple of critical limitations. Firstly, since the spatial information have been lost after down-sampling in the convolution stage, the deconvolution operations are not able to recover the low-level visual features, leading to the inaccurate prediction of high-resolution segmentation outputs. Secondly, the entire network of DENs is nearly twice deeper than FCNs. Training such deeper networks is a nontrivial work, in particular with a limited number of training samples.

In order to overcome these challenges, the context embedding network (CEN) and its variants are proposed to further improve segmentation performance in recent literature [1, 2, 4, 8, 12, 13, 14, 15], which are roughly divided into four categories: skipped-connection, dilated and atrous convolution, CRF-RNN embedding, and cascaded refinement. In the first category, the enhanced version of FCN [1] and hyper-column method [13] encode the context clues for high-resolution estimation, where the mid-layer features are explored using skip connections. Likewise, the DENs, e.g., U-net [12] and Segnet [8], attempt to construct connection between convolutional and deconvolutional network. In contrast, the second category carefully designs convolutional filters to enlarge the re-

*Corresponding author: Quan Zhou, quan.zhou@njupt.edu.cn. This work is partly supported by NSFC (No. 61876093, 61701258, 61762021, 61701252, 61671253), NSFJS (No. BK20181393, BK20170906), NSFGZ (No.[2017]1130, [2017]5726-32), Key Disciplines of Guizhou Province (No. ZDXK[2016]8), and Huawei Innovation Research Program (HIRP2018).

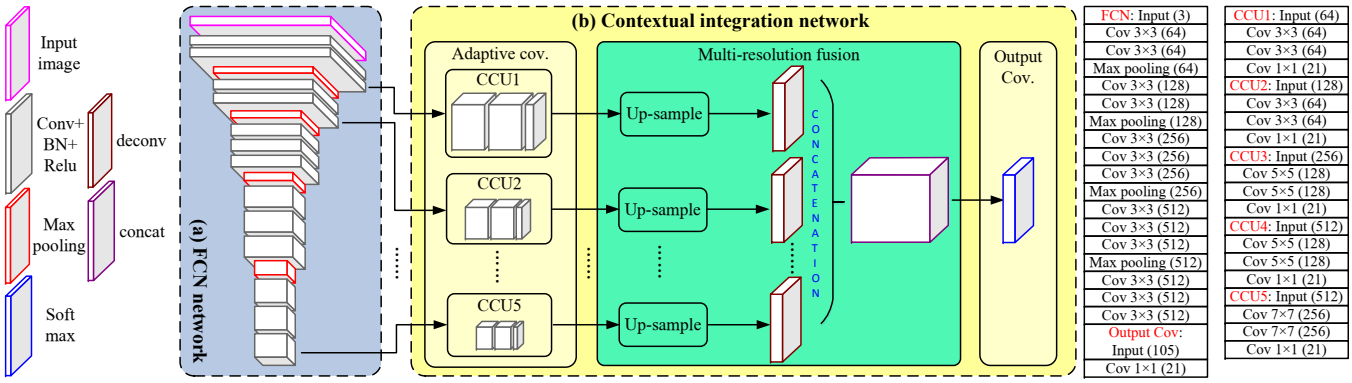

Fig. 1. Overall architecture of the proposed CAN. We integrate convolutional features from all middle layers to abstract context clues, producing the delineated segmentation map of an image. Please refer to text for more details. (Best viewed in color)

ceptive field of network. DeepLab [2] first applies atrous convolution to produce larger size feature maps. Dilated-Net [16] appends several layers after the score map to embed the multi-scale context, while zoom-out [17] proposes a handcrafted hierarchical context features. The representative work of third category include CRF-RNN [14] and deep structured network (DSN) [15], where mean field inference is treated as recurrent layers for end-to-end training of the dense conditional random filed (CRF) and FCN network. For last category, the global context information are captured in a cascaded manner. For example, ParseNet [4] adds a global pooling branch to extract context information. GCNet [3] and ShufferNet [5] carefully design cascaded deconvolution network in score map and feature map, respectively, to capture multi-scale context cues. Although achieving promising results, the previous methods still suffer from the following drawbacks. Firstly, objects tend to be with different scales in image. However, the receptive field of previous networks is not adaptive, leading to the problem that objects substantially larger or smaller than the receptive field may be fragmented or incorrectly classified [2, 7, 8]. Secondly, in spite of using skip connection to investigate context cues, it is hard to integrate all hand-tuned factors or features in an appropriate way. How to conveniently find the optimal context aggregation strategy in FCN framework still remains an open research question in semantic segmentation.

To solve these problems, this paper presents *contextual aggregating network(CAN)*, a generic convolutional feature ensembling framework for semantic segmentation. Motivated by [1, 4], our CAN integrates feature maps from multiple convolutional layers to capture context clues. Features on the top of CAN help the object classification, while shallow layer features contribute to preserve the detailed object shapes and boundaries. As shown in Fig. 1, the FCN-like architecture [1] is employed as backbone network, and CAN is used to generate semantic score maps. Unlike pervious network, we add a new branch, called context convolutional unit (CCU), before each max pooling. Thereafter, these CCUs undergo a multi-resolution fusion block to form multiple scale feature maps, which carry both local and global context. Finally, a output convolution is conducted to fuse different scale context features. Our contributions are three-folds: (1) We introduce a CAN, a generic context aggregation framework for semantic segmentation. Instead of only using several skip connects [1], we argue that features from all intermediate layers are helpful for abstracting context clues. Through refining coarse high-level semantics and fine low-level features, our CAN is able to produce more accurate segmentation outputs. (2) Our CAN can be effectively trained end-to-end, which is crucial for dense labeling problem. Unlike the stage-wised training scheme [1, 2, 14, 15], the simple forward architecture allows us to train CAN in an end-to-end manner. (3) The experimental results show that our CAN outperforms most CEN-based networks on PASCAL VOC 2012 and Cityscapes datasets.

2. OUR METHOD

The entire network of CAN is illustrated in Fig. 1. Note that the architecture of our CAN is generic, where the backbone network (depicted in Fig. 1 (a)) can be replaced by arbitrary FCN-like networks, e.g., VGG-16 [1] or ResNet [5], and the contextual integration network (depicted in Fig. 1 (b)) can be easily modified to accept an arbitrary number of feature maps with arbitrary resolutions and channels.

Network Overview. As shown in Fig. 1, our CAN includes two parts: backbone and contextual integration network. Different parts have different configurations of layers, such as convolution, max pooling, deconvolution and concatenation. The right column of Fig. 1 illustrates the detail structure of backbone and contextual integration network respectively, including layer types, kernel sizes, and the number of channels (in bracket). Unless otherwise stated, the stride typical equals one for convolution, and equals two for max pooling.

The most similar structure to our CAN is skip connection version of FCN [1]. However, three important modifications are required to adapt it to dense estimation problem. Firstly, to

increase the resolution of prediction, we remove the last pooling layer at the top of FCN to enlarge the receptive fields. In this case, we expand the size of network outputs (label maps) by 4 $\times$. In addition, the use of simple structure leads to less network parameters, improving the generalized ability of the whole network. Secondly, to fully investigate multiple scale contextual cues, we fed the convolution layers before max pooling, instead of pooling layers themselves, into our CCUs. Intuitively, compared with pooling layers that have lost spatial information, convolution layers preserve more abundant visual cues, which are helpful for delineating object shapes and boundaries. Finally, as high resolution feature maps consume a large amount of GPU memory in the training process, they limit the size of minibatch (e.g. 8), resulting in the instability of the batch normalization (BN) [18] (as which need to predict sample mean and variance from the training data in a minibatch). We deal with this issue by simply fixing the values of all parameters in BNs, as well as [19] does.

Context convolutional unit. The first part of contextual integration network consists of a series of adaptive convolution that mainly capture multi-scale contextual cues. To this end, each input path is passed sequentially through a context convolutional unit (CCU), which is composed by three adaptive convolutions. To simplify our CAN structure, we adopt a feature reduction in the final convolution. The adaptive convolution helps to re-scale the feature values appropriately along different scales, which is important for the subsequent context fusion. Additionally, as shallow layer of backbone network calculates low-level feature responses, while deeper layer abstracts high-level semantics, we utilize adaptive size of filter kernels in different CCUs. For example, as shown in Fig. 1, the kernel size is gradually increased from 3×3 to 7×7.

Multi-resolution fusion and output convolution. After passing through CCUs, all path inputs are fused into a high-resolution feature map by the multi-resolution fusion block. This block first applies deconvolution to up-sample all (smaller) feature maps to the largest resolution of the inputs, and then fuses all features maps by concatenation. Finally, the stacked feature maps, carrying local and global context, are mapped to a soft-max score map using an output convolution.

3. EXPERIMENTS

3.1. Implementation Details

Dataset. We evaluate our CAN on PASCAL VOC 2012 [20] and Cityscapes [21] datasets, which are popular benchmarks for semantic segmentation. The PASCAL dataset contains 21 object categories (20 foreground categories and one additional background class). Consistently with previous studies, we augment the extra pixel level annotations from [22], which contains 10,582 images for training, 1,449 images for validation, and remaining 1,456 images for testing. The Cityscapes dataset, on the other hand, focuses on street scenes segmentation and includes 19 object categories. Following [2], we only employ images with fine pixel-level annotations, resulting in 2,975 training, 500 validation and 1,525 testing images. The performance is measured in terms of mean pixel intersection-over-union (mIOU) averaged across all classes.

Table 1. Experimental results on Cityscapes test set.

	CRF-RNN	DeepLab	FCN-8s	DPN	DSN	Ours
mIOU	62.5	63.1	65.3	66.8	71.6	71.9

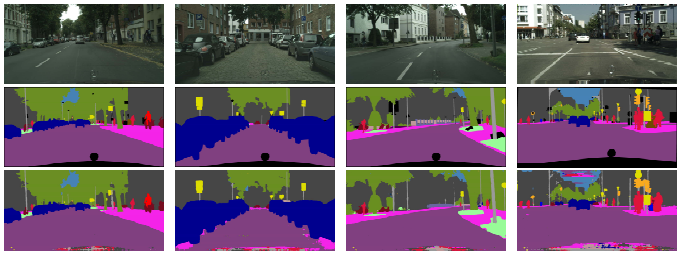

Fig. 2. The visual examples of our method on Cityscapes dataset. From top to bottom are original images, the ground truth, and our predictions. (Best viewed in color)

Baselines. To show the advantages of our CAN, we selected 8 state-of-the-art CEN-based networks as baselines, including FCN-8s [1], Zoom-out [17], DeepLab [2], CRF-RNN [14], LDN [7], DPN [23], SegNet [8], and DSN [15].

Parameter settings. Our entire CAN is implemented on the hardware platform of Intel Xeon E5-2680 server with NVIDIA Tesla K80 GPU, based on Caffe framework [24]. Our CAN is trained using the stochastic gradient descent algorithm [25]. We favor a large minibatch size (set as 8) to make full use of the GPU memory, where initial learning rate, momentum and weight decay are set to 10^{-10}, 0.99 and 0.0005, respectively.

Learning rate policy. Following [2, 9, 26], we employ a "poly" learning rate policy where the initial learning rate is multiplied by $(1 - \frac{iter}{max_iter})^{power}$ with $power = 0.9$.

3.2. Evaluation Results on Pascal VOC

Tab. 2 reports the comparison results between previous methods and our CAN on Pascal VOC 2012 test set. Compared with these CEN-based architecture, our CAN achieves best performance with 76.6% mIOU accuracy, and best scores on 18 out of the 20 categories. Among all baselines, DSN [15] achieves 75.3% mIOU which outperforms other state-of-the-art baselines. The proposed CAN improves the mIOU accuracy by 1.3%. It is intriguing that our approach is superior to the existing methods [2, 7] that employ CRF as post-processing to explore long-range contextual interactions. This indicates our CAN is able to capture wide scale context information, allowing us to estimate more accurate object localizations.

Table 2. Individual category results on the PASCAL VOC 2012 test set in terms of mIOU scores. The bold number indicates the best performance among all approaches for each category.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIOU
SegNet [8]	74.5	30.6	61.4	50.8	49.8	76.2	64.3	69.7	23.8	60.8	54.7	62.0	66.4	70.2	74.1	37.5	63.7	40.6	67.8	53.0	59.1
FCN-8s [1]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [17]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	67.6
DeepLab [2]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [14]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
LDN [7]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
DPN [23]	87.7	**59.4**	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	**62.6**	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
DSN [15]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
Ours	**91.5**	55.0	**88.2**	**69.7**	**75.1**	**93.2**	**86.1**	**87.4**	**40.2**	**82.1**	60.1	**84.6**	**84.2**	**85.9**	**86.4**	**63.3**	**84.4**	**59.2**	**82.0**	**73.4**	**76.6**

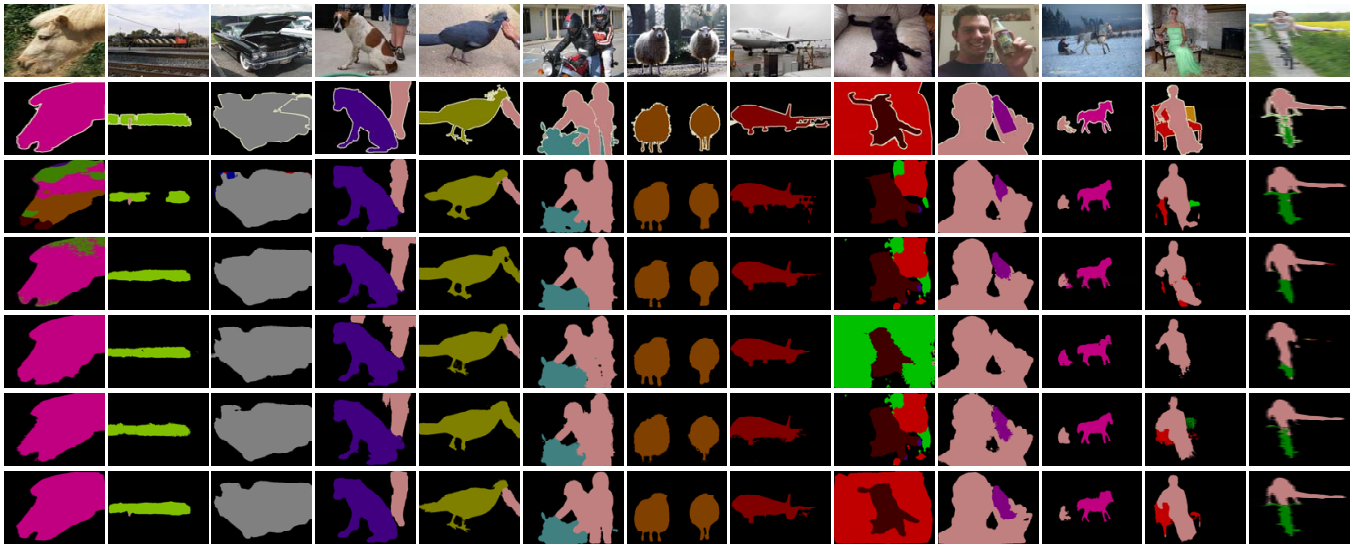

Fig. 3. The visual comparison on PASCAL VOC 2012 val dataset. From top to bottom are original images, the corresponding ground truth, segmentation outputs from FCN-8s [1], DeepLab [2], SegNet [8], LDN [7], and our CAN. (Best viewed in color)

Fig. 3 shows the visual results on the PASCAL VOC 2012 validation set. It is evident that, compared with baselines, our CAN not only correctly classifies object with different scales, but also produces more smooth and detailed segmentation outputs with accurate object shapes and boundaries.

3.3. Evaluation Results on Cityscapes

The images in Cityscapes have a fixed resolution of 1024×2048, which is too large to our network architecture. We thus resize the resolution of images into 256×512 before training stage. We submit our best trained CAN to the online evaluation server, and the comparison detailed results are listed in Tab. 1, which show that CAN outperforms other methods with notable advantage. Using both local and global context makes our CAN yield 71.9% mIOU accuracy. Several visual examples of segmentation outputs are shown in Fig. 2.

4. CONCLUSION AND FUTURE WORK

This paper has described a CAN model, which explores multi-scale context cues for semantic segmentation. Through constructing contextual integration network, our CAN provides a more powerful representation that combines feature maps with different receptive fields. We evaluate our CAN on PASCAL VOC 2012 and Cityscapes datasets. The experimental results show our CAN outperforms recent CEN-based state-of-the-art networks, and demonstrate that our approach can produce more accurate predictions and delineated object boundaries. The future work includes employing more powerful backbone networks, such as [4, 5, 27], and extending our CAN framework for video semantic segmentation tasks.

5. REFERENCES

[1] L. Jonathan, S. Evan, and D. Trevor, "Fully convolutional networks for semantic segmentation," *IEEE TPAMI*, vol. 39, no. 4, pp. 640–651, 2017.

[2] 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.

[3] C. Peng, Z. Xiangyu, Y. Gang, L. Guiming, and S. Jian, "Large kernel matters: Improve semantic segmentation by global convolutional network," in *CVPR*, 2017, pp. 1743–1751.

[4] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Y. Jia, "Pyramid scene parsing network," in *CVPR*, 2016, pp. 6230–6239.

[5] L. Guosheng, M. Anton, S. Chunhua, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017, pp. 5168–5177.

[6] L. Xiaoxiao, L. Zhiwei, L. Ping, L. Chenchange, and T. Xiaoou, "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade," in *CVPR*, 2017, pp. 6459–6468.

[7] N. Hyeonwoo, H. Seunghoon, and H. Bohyung, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015, pp. 1520–1528.

[8] B. Vijay, 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.

[9] W. B. Yang, Q. Zhou, J. N. Lu, X. F. Wu, S. F. Zhang, and L. J. Latecki, "Dense deconvolutional network for semantic segmentation," in *ICIP*, 2018, pp. 1573–1577.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[12] O. Ronneberger, F. Philipp, and B. Thomas, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015, pp. 225–233.

[13] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *CVPR*, 2015, pp. 447–456.

[14] S. Zheng, S. Jayasumana, B. R. Paredes, V. Vineet, Z. Z. Su, D. L. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *ICCV*, 2015, pp. 1529–1537.

[15] G. S. Lin, C. H. Shen, D. H. Van, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE TPAMI*, vol. 40, no. 6, pp. 1352–1366, 2018.

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

[17] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *CVPR*, 2015, pp. 3376–3385.

[18] I. Sergey and S. Christian, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[19] Y. Wang, X. F. Wu, Y. Y. Chang, S. F. Zhang, Q. Zhou, and J. Yan, "Batch normalization: Is learning an adaptive gain and bias necessary?," in *ICMLC*, 2018, pp. 36–40.

[20] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 98–136, 2015.

[21] 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.

[22] B. Hariharan, P. Arbelez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *ICCV*, 2011, pp. 991–998.

[23] 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.

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACMMM*, 2014, pp. 675–678.

[25] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010, pp. 177–186.

[26] W. B. Yang, Q. Zhou, Y. W. Fan, G. W. Gao, S. S. Wu, W. H. Ou, H. M. Lu, J. Cheng, and L. J. Latecki, "Deep context convolutional neural networks for semantic segmentation," in *CCCV*, 2017, pp. 696–704.

[27] H. Kaiming, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2980–2988.