Objectness-aware Semantic Segmentation

Yuhang Wang[1,2], Jing Liu[1], Yong Li[1,2], Junjie Yan[3], Hanqing Lu[1]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]SenseTime Group Limited
{yuhang.wang, jliu, yong.li, luhq}@nlpr.ia.ac.cn, yanjunjie@sensetime.com

ABSTRACT

Recent advances in semantic segmentation are driven by the success of fully convolutional neural network (FCN). However, the coarse label map from the network and the object discrimination ability for semantic segmentation weaken the performance of those FCN-based models. To address these issues, we propose an objectness-aware semantic segmentation framework (OA-Seg) by jointly learning an object proposal network (OPN) and a lightweight deconvolutional neural network (Light-DCNN). First, OPN is learned based on a fully convolutional architecture to simultaneously predict object bounding boxes and their objectness scores. Second, we design a Light-DCNN to provide a finer upsampling way than FCN. The Light-DCNN is constructed with convolutional layers in VGG-net and their mirrored deconvolutional structure, where all fully-connected layers are removed. And hierarchical classification layers are added to multi-scale deconvolutional features to introduce more contextual information for pixel-wise label prediction. Compared with previous works, our approach performs an obvious decrease on model size and convergence time. Thorough evaluations are performed on the PASCAL VOC 2012 benchmark, and our model yields impressive results on its validation data (70.3% mean IoU) and test data (74.1% mean IoU).

Keywords

Semantic Segmentation; Deconvolutional Neural Network

1. INTRODUCTION

Semantic image segmentation is a core problem in computer vision, aiming at parsing images into several semantic regions and assigning them with correct semantic labels. In the past months, tremendous progresses in semantic segmentation have been made based on the framework of fully convolutional neural networks (FCN) [9]. The main advantage of FCN is that the network is an end-to-end network to solve semantic segmentation as a structured pixel-wise labeling problem, i.e., enabling pixel-wise category predictions from a whole image as input.

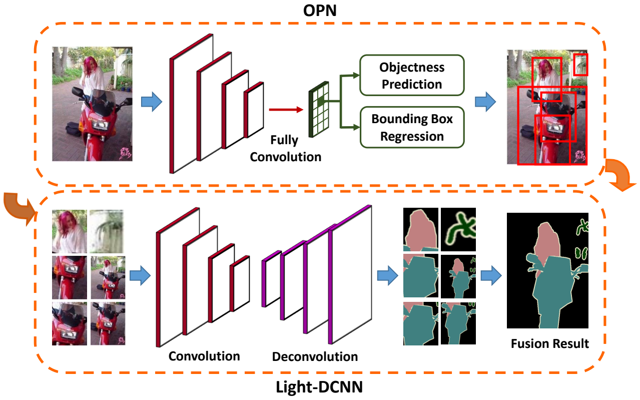

Figure 1: Overall architecture of OA-Seg framework.

However, the pixel-wise supervision in FCN is attached to the upsampled feature maps via large-span bilinear interpolation, thus the object boundaries are over smoothed in the results of segmentation. Besides, given a whole image as input, the fixed-size receptive fields possibly make foreground objects overwhelmed by a large area of diverse background parts. How to overcome the above limitations of FCN is the main motivation of our work. Simply, we aim to find a way to automatically discover region proposals with high objectness scores and segment them by mapping low resolution features to input resolution through a more refined method, where the boundary information is maintained as much as possible.

In view of such a motivation, we design an objectness-aware semantic segmentation framework (OA-Seg), which consists of two CNN-based networks. The first one is an object proposal network (OPN), which is a kind of fully convolutional network and trained end-to-end to generate candidate object regions. OPN takes an image as input and outputs a set of rectangular object proposals associated with their corresponding objectness scores. The second network is a lightweight deconvolutional neural network (Light-DCNN) which decodes convolutional feature maps into the same size of input image, to guarantee finer pixel-wise predictions. Light-DCNN is constructed on the top of convolutional layers in VGG-net with a mirrored deconvolutional and unpooling architecture, while the fully-connected layers are removed. Pooling indices in the convolutional part are employed during unpooling to ensure location consis-

MM '16, October 15-19, 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2964284.2967232

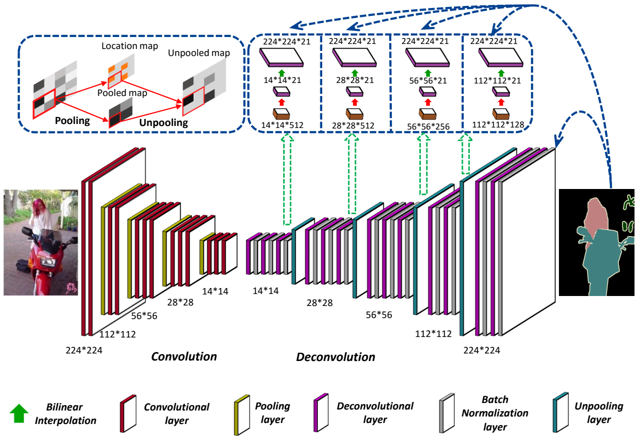

Figure 2: Network structure of Light-DCNN.

tency, which conducts a more refined unsampling operation than bilinear interpolation. Besides, hierarchical classification layers are presented by mapping multi-scale intermediate deconvolutional features to pixel-wise labeling maps, where unpooling and bilinear interpolation are cooperatively used. For the above two networks, we currently train them with object proposals as their connection, however alternative training schemes can also be further adopted to produce a unified network with sharable convolutional layers.

The network we propose above is a basic network for semantic segmentation, like FCN [9] and DeconvNet [11]. While compared with them, our model is much more lightweight with the model size decreased by 3 to 9 times and converges much faster, yet achieves better performance. Other improvements like CRF or MRF, which are always jointly used with FCN in state-of-the-art methods[1, 7, 19, 8], are actually orthogonal to our method. And they can also be combined with our method to further improve the performance. We simply use CRF as post-processing in this paper and evaluate our method on the PASCAL VOC 2012 benchmark [2], where our proposed OA-Seg produces impressive results on its validation data (70.3% mean IoU) and test data (74.1% mean IoU).

2. OUR APPROACH

Our model consists of two jointly learned neural networks as shown in Figure 1. One is an object proposal network (OPN) used to preliminarily locate objects in the image. And the other one is a lightweight deconvolutional neural network (Light-DCNN) for further semantic segmentation. With this framework, we enhance the performance of semantic segmentation with detection techniques, as massive background noises are eliminated during both training and testing.

2.1 Object Proposal Network

OPN is used here to help separating objects from overlapping ones and complex backgrounds, which makes the model more focused on single objects during training and testing. So we use "objectness" as our measurable indicator and make a rough judgment of whether the image region contains an object or not. The network is built on VGG 16-layer net [16] with a fully convolutional structure. All fully connected layers are removed, and the local features of the last convolutional layer are mapped to dense feature vectors through an extra convolutional layer. Thus each feature vector corresponds to a receptive field in the input image. We reuse it to represent multiscale image regions centering around the corresponding receptive field referring to [12], and simultaneously perform classification and bounding box regression to estimate objectness scores and region locations. The groundtruth used here are the circumscribed rectangles of segmentation groundtruth, and thus no extra information is used.

2.2 Light-DCNN

Unlike FCN [9] which obtains pixel-wise prediction through large-span bilinear interpolation, Light-DCNN provides a more refined reconstruction via progressively upsampling the feature maps to larger resolution. Illustration of our network structure is shown in Figure 2.

Light-DCNN is composed of two parts. The first part is a convolutional network that takes an image as input and encodes it into feature maps. And the second part is a deconvolutional network that takes the feature maps as input and propagates the responses back to each pixel. The two networks are spliced and optimized together. For the convolutional network, we directly inherit the network structure and parameters from VGG 16-layer net [16]. The deconvolutional network is built and the parameters are assigned following Zeiler *et al.* [17], which is a reverse process of the convolutional network with symmetrical structure. The main components of the deconvolutional network are deconvolutional and unpooling layers. Unlike convolutional layer that aggregates feature vectors in a local region into a single response, the deconvolutional layer attempts to disassemble the response back to each individual position. The unpooling layer carries out the inverse process of the pooling layer. As we use max pooling in the convolutional network, through the unpooling layer, we upsample the feature maps following max pooling indices and supplement 0 for the abandoned positions to ensure position correspondence.

Besides, our network is designed with several unusual features, which make it small-sized and easy to converge.

Parameter Inheritance: We use the parameters of the learned convolutional layers to initialize the deconvolutional layers in our network, rather than Gaussian random numbers. It supplies our network with applicable initial values during deconvolutional computation. And this treatment makes our network extremely easy to converge.

Discarding Fully Connected Layers: We discard the fully connected layers and reverse the net from the last convolutional layer, because the convolutional layers always show a much better ability for reserving spatial information compared with fully connected layer. And with this design, we decrease the model size by nearly 10 times, which makes our network more lightweight and easier to train because of the fewer parameters.

Asymmetrical Batch Normalization: We employ batch normalization [5] in our network to reduce the internal-covariate-shift during training, but we only add it in the deconvolutional part of Light-DCNN. In our framework, the convolutional network works as a feature extractor which encodes semantic information, and we don't want to break the correlations between the well-learned net layers. While the deconvolutional network is trained as a decoder, so we add a batch normalization layer after each deconvolutional layer for better optimization.

Table 1: Results of hierarchical predictions of Light-DCNN on PASCAL VOC 2012 val set.

Prediction	224×224	112×112	56×56	28×28	14×14
Mean IoU	60.0	62.3	62.8	62.2	60.0

Table 2: Results of OA-Seg with different object proposal settings on PASCAL VOC 2012 val set.

Object Proposal Addition	None	Train	Train&Inference
Mean IoU	63.1	66.7	69.6

Hierarchical Predictions: Although with strict positional correspondence, the unpooling layers tend to break some correlations within image regions for the reason of zero supplementary. To alleviate this, we connect a pixel-wise classification layer to each deconvolutional layer ahead of unpooling layers, and supervise it with pixel-wise groundtruth. In this way, we attempt to guarantee the discrimination and region correlations of our model before every unpooling. And the pixel-wise prediction is realized by feeding the deconvolutional feature maps into a fully convolutional classification layer and upsampling them with bilinear interpolation. It can be found that, with the hierarchical predictions, we are actually cooperatively using unpooling and bilinear interpolation to upsample feature maps to the resolution of input image. That is, the lower layer is attached greater scale change with bilinear interpolation, while the higher layer considers less scale change but more unpooling operations. It also inspires us with that, it is no need to reconstruct the deconvolutional feature maps to exact the same resolution with the input image, which is further discussed in section 3.1.

2.3 Implementation Details

For our current practice, the two networks of OPN and Light-DCNN are trained successively with the object proposals as connection. In our framework, the two networks can also be trained alternatively to share the convolutional layers and integrated into a unified model, as the training scheme of Faster R-CNN [12]. But in this paper, we make the two networks work in a sequential way, i.e., feed the object proposals generated from OPN into Light-DCNN. With this structure, we utilize more multiscale information and more notice is paid on tiny objects, which benefit the performance of our method.

In the training stage, the object proposals are selected to train Light-DCNN and those with little overlapping with groundtruth or oversized aspect ratios are discarded. While during testing, only the top n proposals with the highest objectness scores are input into Light-DCNN, as well as the entire image. Then we average the corresponding predictions for each pixel and achieve the final segmentation result. In this paper, we set n to 20.

Both of our two networks are implemented with Caffe [6]. For the training of OPN, the input image is rescaled to 600 pixels on its shorter side, in order to maintain the saliency of tiny objects. And nonmaximum suppression (NMS) is employed during inference to reduce highly overlapped proposals. While for Light-DCNN, we set a small batchsize of 1 in our experiment and average the gradients over all the pixels in a training batch for back propagation. The initial learning rate, momentum and weight decay used for standard stochastic gradient descent (SGD) are set to 2.5e-5, 0.99 and 0.0005 respectively.

3. EXPERIMENTS

Our experiments are conducted on the PASCAL VOC 2012 dataset [2] with extended annotations from [3], which contains 20 object categories and one background category. The extended dataset contains 10582 training images, 1449 validation images and 1456 testing images. We verify our approach on both the validation and test set with mean intersection-over-union (mean IoU) as our metrics.

3.1 Evaluation of Light-DCNN

We firstly evaluate the performance of Light-DCNN individually in this section. The network is trained and tested directly on entire images with no object proposals used.

- **Performance of Hierarchical Predictions**

Light-DCNN is connected to hierarchical classification layers by cooperatively using unpooling and bilinear interpolation, as discussed in section 2.2. If we reconstruct the network up to the resolution of input images, we will get 5 segmentation predictions from different deconvolutional layers with ×16, ×8, ×4, ×2, ×1 bilinear interpolation respectively, while the upsampling ratio through unpooling is just inverse. Thus, given an image of size 224×224, we indicate each of the 5 segmentation predictions with the sizes of their corresponding deconvolutional feature maps, which are 14×14, 28×28, 56×56, 112×112, and 224×224. The performance for each of them is listed in Table 1. It can be found that, the best result appears in the 56×56 layer. While the 224×224 layer, which is upsampled to the input image resolution all by unpooling, achieves poorer performance. So as to the 14×14 layer which is upsampled only with bilinear interpolation. The results proves the effectiveness of cooperatively using unpooling and bilinear interpolation. As two different upsampling methods, bilinear interpolation reserves local consistency but is easy to over smooth object edges, while unpooling restores more refined contexture features but tends to break up the correlations within local regions. The result of 56×56 layer benefits from appropriate proportions for each of them, which is achieved through two unpooling layers followed by ×4 bilinear interpolation.

Therefore, in the following experiments, we train our model with the deconvolutional network built only to the 56×56 layer and remove the upper larger-scale layers, yet the lower hierarchical supervision is still reserved. And only the result of the 56×56 layer is reported, which is the last output of the network.

- **Performance with Different Image Scales**

Considering the influence of image scales on training and inference, we reproduce our experiments with images rescaled to 224×224 and 512×512 respectively, with the network structure mentioned above. The results are shown in Table 5. The performance of our model is improved from 63.1% to 63.9% with larger images, yet with a cost of more training time and computing resource. On consideration of this, the following experiments are all conducted with the input scale of Light-DCNN set to 224×224.

3.2 Evaluation of OA-Seg

In this section, we accomplish the entire process of the proposed OA-Seg framework. During both training and infer-

Table 3: **Comparison with state-of-the-art methods on PASCAL VOC 2012 val set.**

Method	DeconvNet[11]	Zoom-out[10]	DeepLab[1]	Piecewise[7]	Deep-struct[13]	DPN[8]	OA-Seg	OA-Seg+CRF
Mean IOU	67.1	69.9	67.6	**70.3**	64.1	67.8	69.6	**70.3**

Table 4: **Comparison with state-of-the-art methods on PASCAL VOC 2012 test set.**

Method	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mean
FCN-8s [9]	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
DeconvNet [11]	85.9	42.6	78.9	62.5	66.6	87.4	77.8	79.5	26.3	73.4	60.2	70.8	76.5	79.6	77.7	58.2	77.4	52.9	75.2	59.8	69.6
Zoom-out [10]	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	69.6
Piecewise [7]	87.5	37.7	75.8	57.4	72.3	88.4	82.6	80.0	**33.4**	71.5	55.0	79.3	78.4	81.3	**82.7**	56.1	79.8	48.6	77.1	66.3	70.7
DeepLab [1]	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
RNN [19]	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
DPN [8]	**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**
OA-Seg	83.3	44.3	85.3	58.6	**73.4**	91.1	83.6	85.4	31.6	78.5	61.0	78.4	80.9	84.3	80.4	61.8	85.3	52.8	82.3	66.7	73.5
OA-Seg+CRF	83.9	45.4	**86.9**	59.0	73.1	**91.6**	**83.8**	**86.5**	31.7	79.5	61.0	79.2	**81.4**	**85.3**	81.7	**63.3**	**86.3**	**53.4**	**82.8**	66.7	**74.1**

Table 5: **Analysis on Model Size and Convergence on PASCAL VOC 2012 val set.**

Method	Input Scale	Convergence Time	Model Size	Mean IOU
FCN-8s [9]	500	120 hours	513M	-
DeconvNet [11]	224	168 hours	961M	67.1
light-DCNN-224	224	11 hours	114M	63.1
light-DCNN-512	512	37 hours	114M	63.9
OA-Seg	224	77 hours	180M	**69.6**

ence, the performance of our model is improved with object proposals added. And state-of-the-art results are achieved on both the validation and test set.

• Performance with Different Object Proposal Settings

We verify the effectiveness of our proposed OA-Seg framework by progressively adding object proposals to training and inference processes. The results are shown in Table 2.

Firstly, we add object proposals only for training and evaluate on entire images, the performance of our model improves obviously by 3.6 percent. While with object proposals added to the testing process as well, our result achieves a further improvement by 2.9 percent, which verifies the effectiveness of OA-Seg. With object proposals provided, more focused samples are supplied during both training and inference and too large or tiny objects are rescaled to suitable sizes, which help improve the discrimination of our model.

• Comparison with State-of-the-art Methods

Performance of our proposed OA-Seg is compared with some of the best methods on validation and test sets, as shown in Table 3 and 4 [1]. We achieve state-of-the-art results with 70.3% mean IoU accuracy on validation set and 74.1% mean IoU accuracy on test set, with best results on 10 categories.

It should be noticed that, actually only FCN [9] and Deconvnet [11] can be directly compared with our model as basic segmentation networks. And our model obviously outperforms them by 11.3 percent and 3.9 percent respectively on test set, without the post-processing of CRF.

When combined with CRF/MRF, the FCN-based methods show improvements with CRF used just as post-processing [1]. And the results are further improved when jointly training CRF/MRF with FCN [7][19][8]. In this paper, we simply use CRF as post-processing and achieve state-of-the-art results on both the validation and test sets.

3.3 Analysis on Model Size and Convergence

We provide more indexes of our model in this section for comprehensive comparison with FCN [9] and DeconvNet [11], as all these three models work as basic networks in semantic segmentation. The detailed comparison is shown in Table 5. For our approach, we report the performance of Light-DCNN with image sizes of 224 and 512. Moreover, the performance of complete OA-Seg is reported with the input scale of Light-DCNN set to 224.

Under each of the settings, our approach always shows much faster convergence rate. With Light-DCNN trained with 224×224 images, we already achieve a result comparative with or better than FCN, referring to its test set result 62.2% [9]. But the convergence time is reduced more than 10 times. If we employ a similar image scale with FCN, our result further improves. When we accomplish the entire framework of OA-seg, the time cost increases to 77 hours which is still about half of the other two methods, yet the mean IoU result of our model obviously exceeds theirs.

Moreover, we decrease the model size by about 3 to 9 times compared with FCN and DeconvNet, which confirms that our models is more lightweight, flexible-to-use and effective in practice.

4. CONCLUSION

We propose an objectness-aware semantic segmentation framework in this paper, which consists of OPN and Light-DCNN. OPN is used to generate object proposals and make our model more focused on objects. While Light-DCNN provides more refined reconstruction to pixel-wise predictions and meanwhile lightens the network. The two networks are learned jointly to enhance the performance of semantic segmentation. We achieve impressive results on PASCAL VOC 2012 dataset with significant reduction on model size and convergence time, which confirms our approach as an effective and efficient framework for semantic segmentation.

5. ACKNOWLEDGMENTS

This work was supported by 863 Program (2014AA015104) and National Natural Science Foundation of China (61332016, 61272329, and 61472422).

[1] The anonymous results links:
http://host.robots.ox.ac.uk:8080/anonymous/LGAA2D.html
http://host.robots.ox.ac.uk:8080/anonymous/ARIGY7.html

6. REFERENCES

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

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[3] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011.

[4] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 2015.

[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[7] G. Lin, C. Shen, I. Reid, and A. van den Hengel. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

[8] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015.

[9] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[10] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[11] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[12] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[13] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

[14] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages 1–15. 2006.

[15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[17] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. 2014.

[18] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pages 2018–2025, 2011.

[19] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.