OBJECT SEGMENTATION WITH DEEP REGRESSION

Jianchao Li[†‡] *Dan Wang*[‡†] *Canxiang Yan*[*] *Shiguang Shan*[†]

[†]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[‡]University of Chinese Academy of Sciences, Beijing, 100049, China
[*]Institute of Deep Learning, Baidu, Inc., Beijing, 100085, China
{jianchao.li, dan.wang}@vipl.ict.ac.cn, yancanxiang@baidu.com, sgshan@ict.ac.cn

ABSTRACT

Object segmentation has constantly received much attention due to its fundamental role in scene understanding. Traditional methods formulate it as a structured prediction problem, represented by graphical models (GMs). However, most GMs have difficulties in balancing the effectiveness of context modeling and efficiency of model inference. In this paper, we model the contexts implicitly using the deep convolutional neural network (DCNN). Specifically, we reformulate object segmentation as a regression problem and train a deep network end-to-end to learn the nonlinear mapping from the image to the object mask. The large receptive field of the network incorporates wide contexts to update the network parameters, giving an implicit context model. Moreover, the deep architecture is favorable for modeling nonlinearity. The inference of our method is quite efficient, involving only a simple feed-forward pass. Extensive experiments on public datasets demonstrate the advantages of our method.

Index Terms— Object segmentation, context modeling, deep convolutional neural networks, regression

1. INTRODUCTION

Object segmentation, an important problem in vision areas, is typically formulated as a structured prediction problem which requires us to predict the labels $\mathbf{y} = \{y_1, \ldots, y_m\} \in \{0, 1\}^m$ (*i.e.*, 0 for backgrounds and 1 for objects) of m pixels/superpixels given their features $\mathbf{x}$. The main difficulty in this formulation is how to effectively model the complex relationships (contexts) within $\mathbf{y}$, and, at the mean time, efficiently infer the model [1].

Conditional Random Fields (CRFs) [2] boil the problem down to minimizing an energy function, with high-order potentials for modeling contexts. Though higher-order potentials model wider contexts, minimization with them is non-trivial. Consequently, common CRFs involve only up to second-order (pairwise) terms, which simplifies the inference but limits the performance since many contexts are left unmodeled. Some researchers try to design higher-order potentials with nice properties that account for more contexts and enable tractable inference [3, 4]. Their work improves greatly over pairwise CRFs and inspires others to seek for better models for contexts.

Recently, Restricted Boltzmann Machines (RBMs), a type of shallow neural network, are introduced to segmentation [5, 1]. RBMs model the relationships within $\mathbf{y}$ via hidden variables $\mathbf{h}$. Their power for this task has also been proven [5, 1]. However, the shallow architecture of RBMs may still be insufficient for modeling highly nonlinear contextual relationships, which calls for a deep architecture.

In this paper, we employ deep convolutional neural networks (DCNNs) for context modeling. Deep networks have recently boosted the performance of many vision tasks such as image classification [6, 7], object detection [8], facial point detection [9] and pose estimation [10]. In fact, they are also suitable for object segmentation. Their deep architecture leads to a wide receptive field, which incorporates a wide range of context information into the output. These information is then exploited to update the network by backpropagating the output error, implicitly modeling the contexts. Moreover, their feed-forward nature allows for efficient inference, which may overcome the limitations of GMs.

In fact, many researchers have applied DCNNs to segmentation. However, the deep networks are more suitable for single-label prediction instead of the dense prediction required by segmentation. Consequently, most researchers adopt the deep networks only as a component, mostly as a feature extractor by treating output of intermediate layers as features, in their frameworks. In contrast, we employ the networks in an end-to-end manner, taking images as input and generating the object masks through the forward pass. Experimental comparisons with state-of-the-art methods on public datasets demonstrate the advantages of our method.

2. RELATED WORK

CRF-based methods. To overcome the limitations of pairwise CRFs, some researchers design novel higher-order po-

tentials. Kohli *et al.* [3] propose a robust higher order potential on image segments to model region consistency. Shekhovtsov *et al.* [4] learn a curvature (local contexts) prior for the object through encoding higher-order potentials as lower envelopes of linear functions. These higher-order terms improve greatly over the pairwise one, with some structural assumptions to allow for tractable inference. However, these explicit context models still confront quite complex inference. Motivated by [11], we seek for implicit context models that may bring higher efficiency.

RBM-based methods. Li *et al.* [5] use a one-layer RBM to learn the *Compositional High Order Pattern Potentials* (CHOPPs) while Kae *et al.* [12] introduce a virtual pooling layer to an RBM to model global shapes (global contexts). Similarly, Yang *et al.* [1] present a *Max-Margin Boltzmann Machine* (MMBM) that adds one layer between $\mathbf{x}$ (features) and $\mathbf{h}$ (hidden variables) for shape prediction. However, all these shallow architectures may be insufficient for modeling the highly nonlinear contextual relationships, for which a deep architecture may be more favorable.

DCNN-based methods. Farabet *et al.* [13] extract multi-scale features of each pixel in a patch-wise manner using a ConvNet [14] and then classify the pixels into different semantic regions using softmax regression, followed by some post-refinement. Sharma *et al.* [15] adopt a similar approach and propose a *recursive Context Propagation Network* (rCPN) to refine the initial pixel classification results. The idea of Mostajabi *et al.* [11] is also analogous except they choose shallow neural networks to classify superpixels after extracting their features by DCNNs. All these work achieved impressive results due to the great expressive power of features learned by deep networks [16]. Our work distinguishes from theirs in that we keep in line with the end-to-end learning philosophy that has driven deep learning [11]. Specifically, we feed images to the network and obtain object masks through the feed-forward pass.

3. THE METHOD

In our method, the deep network is trained end-to-end to learn the nonlinear mapping form the image to the object mask. In fact, the network can be viewed as an implicit context model. In the forward pass, a wide range of context information is incorporated into the output, whose error is then backpropagated to update the network parameters. Thus the contexts implicitly guide the training process, after which an implicit context model is learned. In addition, the deep architecture itself is a favorable model for the nonlinear relationships between images and object masks.

3.1. Implicit Context Modeling

A typical deep convolutional network is composed of several types of layers, *i.e.*, convolutional, pooling and fully-connected layers. Each layer of data is a four-dimensional *blob* of size $h \times w \times c \times n$, which means that it contains n arrays of height h, width w, and number of channels c. In the forward pass, the context information in a single array is incorporated layer by layer into the output.

The convolutional layer applies a set of convolutional kernels to each array. Each output unit of the convolutional layer is related to a window (of the same size as the convolutional kernel) of input units, which incorporate the context information within that window. For example, applying an $l \times l$ kernel $\mathbf{K}$ to a single-channel array $\mathbf{A}_{\mathrm{i}}$ of size $h \times w$ without padding gives an output array $\mathbf{A}_{\mathrm{o}}$ of size $(h-l+1) \times (w-l+1)$, whose (j,k)-th element is

$$\mathbf{A}_{\mathrm{o}}(j,k) = \sum_{p=1}^{l}\sum_{q=1}^{l} \mathbf{K}(p,q)\mathbf{A}_{\mathrm{i}}(j+p-1, k+q-1).$$

Obviously, each output element is related to a $l \times l$ window of input elements, incorporating their contexts. The pooling layer replaces the outputs in a window with a statistical metric (*i.e.*, maximum for max-pooling and mean for mean-pooling), which in fact enforces local smoothness constraints in that window. The fully-connected layer, parameterized by a weight matrix $\mathbf{W}$ and a bias vector $\mathbf{b}$, maps input $\mathbf{t}$ to output $\mathbf{s}$ via $\mathbf{s} = \mathbf{W}\mathbf{t} + \mathbf{b}$. In this case, the context window contains all the input units, resulting in a large receptive filed that incorporates considerably wide contexts.

After the forward pass, the output has incorporated many contexts from the input. Then the output error is backpropagated to update the network, which implicitly utilizes the context information for parameter adjustment. Consequently, the learning phase of the network can just be viewed as an implicit modeling process of the context information.

3.2. Deep Regression

Inspired by the implicit context modeling process in deep networks, we apply them to segmentation. Unlike most existing methods [13, 15, 11], we keep the end-to-end learning manner of the network, which enables efficient inference.

Denoting the image and the object mask as $\mathcal{I}$ and $\mathcal{S}$ respectively, we reformulate the segmentation task as a regression problem, which requires us to learn a mapping $\mathbf{F}$ from $\mathcal{I}$ to $\mathcal{S}$:

$$\mathbf{F} : \mathcal{I} \longrightarrow \mathcal{S}. \tag{1}$$

Generally, $\mathbf{F}$ is highly nonlinear, which makes the deep network a natural choice of model. In fact, the deep architecture of the network usually presents a highly nonlinear relationship between the input and the output. The explicit nonlinearity layers (such as ReLU [17] layers) within the network also help to increase the degree of nonlinearity of the learned mapping. Specifically, the segmentation task is formulated as the following optimization problem:

$$\min_{\mathbf{F}} ||\mathbf{F}(\mathcal{I}) - \mathcal{S}||_2^2, \tag{2}$$

where $\mathbf{F}$ is parameterized by weights $\mathcal{W}$ and biases $\mathcal{B}$ (weights and biases of all layers of the network). Moreover, we include a regularization term (a weight decay term) $||\mathcal{W}||_F^2$, which penalizes weights of large magnitudes, in the optimization objective to prevent overfitting. Eventually, the segmentation task is formulated as:

$$\mathbf{F}^* = \arg\min_{\mathbf{F}} \left(||\mathbf{F}(\mathcal{I}) - \mathcal{S}||_2^2 + \lambda ||\mathcal{W}||_F^2 \right), \qquad (3)$$

in which λ (called the weight decay factor) controls the relative importance of the regularization term.

During training, the network solves Equation (3) to learn the nonlinear mapping $\mathbf{F}^*$ that maps an image to the object mask. The inference then involves only a feed-forward pass, which is quite efficient.

4. EXPERIMENTS

4.1. Implementations

Network. We employ an off-the-shelf deep network — CaffeNet, offered by `Caffe` [18]. This network is a variant of the ILSVRC12 winner AlexNet [7]. It contains 5 convolutional layers (3 of them followed by pooling layers) and 3 fully-connected layers. The original output layer `fc8` generates a 1000-dimensional vector, with elements being the probabilities of the image belonging to the 1000 classes of ImageNet [6]. We change the number of output units to be the same as number of pixels in the object mask. Figure 1 shows a network used in our experiments. The `fc8` layer outputs a long vector (the last long thin rectangle), which is then reshaped to the desired object mask.

Training. We use the parameters of CaffeNet trained on ImageNet for initialization. Then we finetune the network by solving Equation (3) using stochastic gradient descent with momentum. We use the same momentum value 0.9 for all the experiments and tune other parameters (*i.e.*, the learning rate and the weight decay factor λ) on a validation set. The finetuning is performed on a Cuda Tesla K40 GPU.

Data Augmentation. Training a deep network requires a large amount of data, otherwise the network may overfit and become unstable. However, number of images in most segmentation datasets is relatively small due to the great expenses of obtaining pixel-wise labeled data. Consequently, it is necessary to perform data augmentation. For each training image, we perturb it with differences in rotation, cropping, scaling and flipping. Finally, each training image will be augmented with 30 images.

Test. For tests, we use the MATLAB wrapper of `Caffe`.

Refinement. The object mask can be further refined by explicit context models, such as Graph Cut [19]. We adopt the refinement method in [1], using the mask value as the unary term and employing the pairwise term from [19].

4.2. Datasets

Penn-Fudan Pedestrian Database [20] contains 170 images, each with one to seven pedestrians. Both a bounding box and a pixel-wise annotation is provided for each pedestrian. We crop them out using the bounding box and include their mirror, resulting in 846 images. Then we split them into 400 training images and 446 test images, while guaranteeing an image and its mirror appear together in either the training or the test split.

Weizmann Horse Database [21] contains 328 images, each with exactly one horse. We split them into 200 training images and 128 test images. In addition, the ground truth is resized to 32×32 for evaluation as in [5] and [1].

Caltech-UCSD Birds 200 [22] is much larger in scale than previous two, with 6033 images each containing one bird. Accurate segmentation annotations are provided by [1]. We crop each bird out using its bounding box and adopt the training/test split of [22], (into 3000 training images and 3033 test images). Moreover, we resize the cropped ground truth to 128×128 for evaluation as in [1].

In fact, we strictly follow the cropping, training/test splitting and evaluation strategies of [1] for fair performance comparisons in 4.3.

After splitting the data, we leave out one fifth of images in the training split as the validation set for parameter tunings (data augmentation is only applied to the remaining four fifths of images in the training split). Finally, the learning rates are set to 10^{-5} for the pedestrian and bird datasets and 10^{-4} for the horse dataset while the weight decay factor λ is fixed to 10^{-3}. Moreover, we remove the `fc8` layer for the pedestrian dataset after observing overfitting by using the full CaffeNet architecture.

In addition, we resize the ground truth object mask to a smaller size for training. After inference, we resize the output mask back to the original size for evaluation. Specifically, we resize the ground truth masks of pedestrians and horses to 32×32 and birds to 64×64.

4.3. Results

Metrics. We report three metrics for performance comparisons with previous state-of-the-art methods, that are the average pixel accuracy (APA) and the Intersection-over-Union (IoU) score for measuring effectiveness, and the average running time (Time) for measuring efficiency. All the tests are performed on a desktop machine with an Intel Core i7-3770 CPU (3.40 GHz). We use the codes of [5] and [1] to record the time of their methods. Moreover, in Tables 1 to 3, we name our method *DeepReg* because it uses *Deep* networks to perform *Reg*ression.

Results on Penn-Fudan Pedestrian Database. Table 1 shows the results on the pedestrian dataset, which seems to be challenging for DeepReg. It performs slightly better than CHOPPs, which only uses a one-layer RBM for global

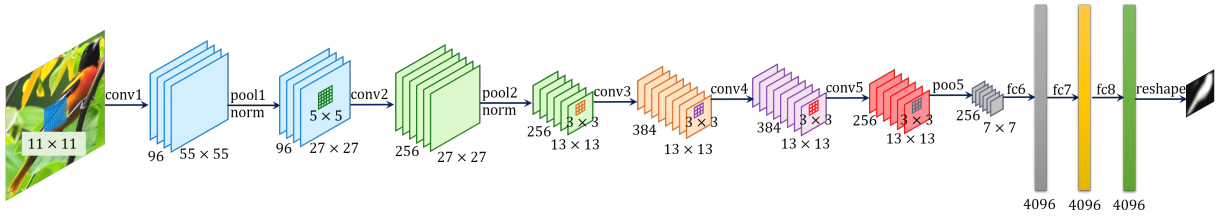

Fig. 1. A network used in our experiments, taking an image (leftmost) as input and generating a long vector (the last long thin rectangle) after the `fc8` layer as output, which can be reshaped to the object mask. Note that the object mask is resized to a smaller size for regression, as shown above.

	APA (%)	IoU (%)	Time (s)
CHOPPs [5]	86.55	71.33	N/A
MMBM [1]	89.91	76.92	19.01
MMBM+GC [1]	**90.77**	**79.42**	19.06
DeepReg	87.58	72.94	1.08
DeepReg+GC	89.65	76.93	1.13

Table 1. Results on the Penn-Fudan Pedestrian Database.

	APA (%)	IoU (%)	Time (s)
CHOPPs [5]	88.67	69.90	1.34
MMBM [1]	89.80	72.09	N/A
MMBM+GC [1]	90.71	75.78	N/A
DeepReg	92.87	74.17	1.01
DeepReg+GC	**94.91**	**80.44**	1.16

Table 2. Results on the Weizmann Horse Database.

context modeling. However, it is are inferior to MMBM (a two-layer RBM). We analyze the bad cases of DeepReg on this dataset and find most of them contain one target pedestrian in the middle with several in surrounding backgrounds. And DeepReg gives strong responses to all of them instead of just the target, which lowers its performance. This is possibly due to the difficulty for an implicit context model to differentiate the target object from those of the same class. In fact, DeepReg performs the best on both the following datasets that do not present such phenomenon, which concurs with this explanation.

Results on Weizmann Horse Database. Table 2 shows the results on the horse dataset, on which DeepReg performs slightly the best. During Graph Cut, we resize the 32×32 object masks predicted by DeepReg back to the sizes of the original images for refinement. These larger images contain more context information in higher resolution, which helps Graph Cut to achieve large improvements.

Results on Caltech-UCSD Birds 200. DeepReg significantly outperforms the state-of-the-art on this dataset, as shown in Table 3. The large amount of training data in this set enables full finetuning of the deep architecture, which effectively models the nonlinear contextual relationships between images and object masks.

	APA (%)	IoU (%)	Time (s)
CHOPPs [5]	74.52	48.84	N/A
MMBM [1]	88.07	72.96	12.21
MMBM+GC [1]	90.42	75.92	12.27
DeepReg	90.66	80.61	1.04
DeepReg+GC	**91.80**	**82.59**	1.10

Table 3. Results on the Caltech-UCSD Birds 200.

Moreover, DeepReg runs much faster than GM-based methods due to the efficient inference of deep networks. MMBM takes significantly more time because it involves the computation of the Ultrametric Contour Map [23], which is quite time-consuming.

5. CONCLUSIONS

GM-based methods for segmentation usually confront the conflict between effectiveness of context modeling and efficiency of model inference. In this paper, we adopt deep convolutional neural networks as implicit context models. The network incorporates increasingly wider contexts layer by layer into the output, whose error is backpropagated to update its parameters. The inference process is quite efficient, involving only a simple feed-forward pass. We then reformulate segmentation as regression and train a deep network end-to-end to learn the nonlinear mapping from images to object masks. The deep architecture of the network also makes it desirable for the high nonlinearity of the mapping.

Though the implicit context model has difficulties in distinguishing between multiple target-class objects present together, it achieves state-of-the-art performance on two public datasets with more efficient inference, showing the potential to resolve the above conflict.

6. ACKNOWLEDGEMENTS

This work is partially supported by 973 Program under contract No. 2015CB351802; Natural Science Foundation of China under contracts Nos. 61222211, 61390511, and 61402430; and China Postdoctoral Science Foundation.

7. REFERENCES

[1] J. Yang, S. Safar, and M.-H. Yang, "Max-Margin Boltzmann Machines for Object Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[2] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *The 18th International Conference on Machine Learning*, 2001.

[3] P. Kohli, L. Ladický, and P. H. Torr, "Robust Higher Order Potentials for Enforcing Label Consistency," *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, May 2009.

[4] A. Shekhovtsov, P. Kohli, and C. Rother, "Curvature Prior for MRF-Based Segmentation and Shape Inpainting," in *Pattern Recognition*, vol. 7476 of *Lecture Notes in Computer Science*, pp. 41–51. 2012.

[5] Y. Li, D. Tarlow, and R. Zemel, "Exploring Compositional High Order Pattern Potentials for Structured Output Learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and F.-F. Li, "ImageNet Large Scale Visual Recognition Challenge," *arXiv preprint arXiv:1409.0575*, 2014.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Informantion Processing Systems Conference*. 2012.

[8] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[9] Y. Sun, X. Wang, and X. Tang, "Deep Convolutional Network Cascade for Facial Point Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[10] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," *arXiv preprint arXiv:1412.0774*, 2014.

[12] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, "Augmenting CRFs with Boltzmann Machine Shape Priors for Image Labeling," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning Hierarchical Features for Scene Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, August 2013.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in *Proceedings of the IEEE*, pp. 2278–2324. 1998.

[15] A. Sharma, O. Tuzel, and M.-Y. Liu, "Recursive Context Propagation Network for Semantic Scene Labeling," in *Advances in Neural Informantion Processing Systems Conference*. 2014.

[16] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision*, 2014.

[17] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *ACM International Conference on Multimedia*, 2014.

[19] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," in *International Conference on Computer Vision*, 2001.

[20] L. Wang, J. Shi, G. Song, and I.-F. Shen, "Object Detection Combining Recognition and Segmentation," in *Asian Conference on Computer Vision*, 2007.

[21] E. Borenstein and S. Ullman, "Class-Specific, Top-Down Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2002.

[22] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.

[23] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011.