High-Accurate Segmentation of Shape Constrained Object

Anonymous ICCV submission

Paper ID ****

Abstract

Object segmentation has recently made great progress due to powerful features extracted using deep convolutional neural networks (CNNs). However in many practical scenarios, prior shape constraint about segmented object is usually available. Incorporating such prior knowledge into CNNs architecture to obtain the accurate morphological shape is desirable, such as vesicle segmentation. In this paper, we propose an effective shape constrained network (scnet) to constrain all the components of a segmentation to satisfy the prior knowledge. Specifically, a divide-and-conquer strategy is adopted by breaking down the segmentation into object orientation and outline depiction subtask, which is analogy to RPN in object detection. Furthermore a joint max pooling operation is developed to fuse the orientation and outline results into a compact form, which can be directly transformed to the segmentation result. The whole process is trained end-to-end and residual error can be correctly propagated. Our proposed method is generic as it is further used for object detection task by treating detection bounding box as rectangular object to be segmented. Experimental results will prove the effectivity and general of our method. Code is made publicly available at http://www.pamitc.org/documents/mermin.pdf..

1. Introduction

Image segmentation is a challenging task which aims at segmenting the objects from complex background. In practical application, it often occurs that there are many prior constraints on object shape, such as elliptical cell, polygonal bricks and rectangular tiles. Especially for biological image, most membrane structure in one Electron microscopy (SEM) images have a similar shape. For example, a typical presynaptic structure contains dense vesicles of different type, which is crucial for estimating the synaptic activity [6][5]. Accurate segmentation of these vesicles is a crucial pre-requisite step to obtain reliable morphological statistics, including vesicle position, inclination angle, length of major axis.

Nevertheless, this task is quit challenging for several reasons. First, the objects such as vesicle or bricks usually get dense together, which make it easily suffer from serious touching problem [3], as shown in Fig. Second, since the shape of objects is usually concise and compact, demand on boundary region segmentation is much higher, which is more difficult for learning. Third, exact morphological statistics of objects are desired to be directly obtained from the network, instead of additional manual measuring in conventional strategy.

Most existing segmentation methods using deep convolutional neural networks (CNNs) [4],[17],[1],[3],[18] to predict a label for each pixel in an image. However they do not model the interactions between output variables directly, thus the boundaries of segmentation object are usually coarse and smooth, which is terrible for some object with sharp outline shape. Moreover without any shape constraint on segmenting objects, contiguous objects are easily touching each other, as shown in Fig. [2] solves this problem by using edge prediction to cleave the touching cells, while [11] add the loss weights around the boundary regions. However the boundaries obtained by cleaving is coarse and exact morphological statistics still needs further operation. Some methods for segmenting specific shape structure are not deep neural architecture, which commonly segment multiple structures sequentially and can't be globally optimized among the whole input image. [8],[14],[12],[16],[15],[7]. Few study has successfully incorporating these shape constraint into the deep convolutional neural networks (CNNs) as an unified framework.

In this work, we propose the first shape constraint network (scnet) to segment dense objects with certain shape constraint, while simultaneously resulting in corresponding morphological statistics for each object. We formulate the shape constraint as a set of parameters, which can depict the outline shape of object. Analogy to Region Proposal Network in [10], the previous part of our scnet produce a set of parameterized shape depiction, each with an objectness score The insight is to divide this challenging task into two easier part, one designed for finding out all the possible objectness region, another one designed for depicting

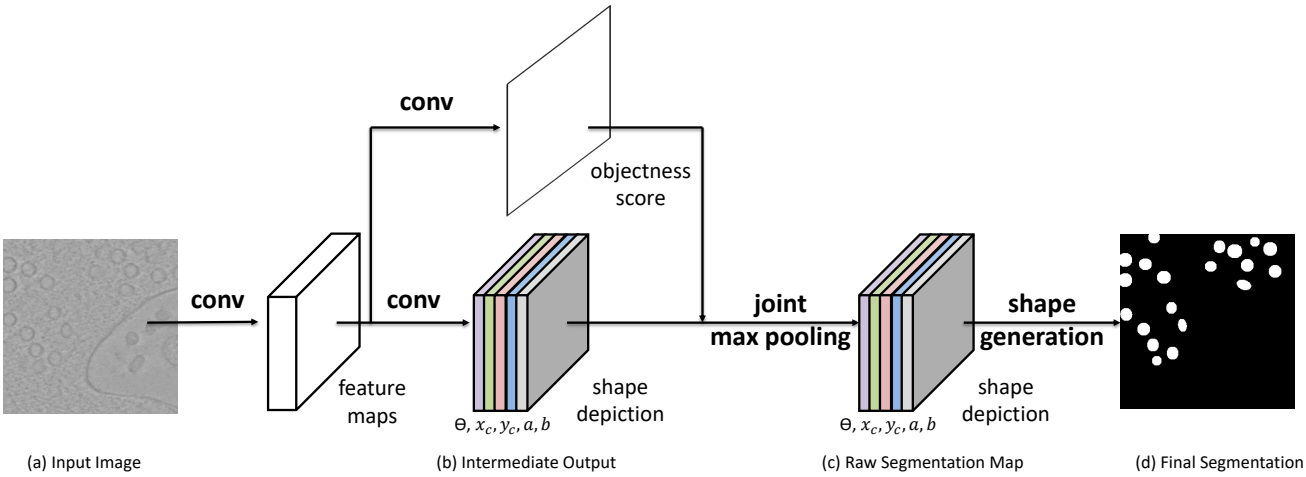

Figure 1. Overview of our proposed scnet. Given an input image (a), shape constraint net first use deeplab to get the feature map and generate two intermediate outputs (b) including objectness score and shape depiction. Then the joint max pooling operation is applied to fuse (b) into a raw segmentation map, which is finally transformed to final segmentation (d) by shape generation layer. The whole network can be jointly trained end-ro-end.

the ourline shape of possible object. In order to make our scnet effectively using these two abilities, a joint max pooling operation was developed to fuse above two outputs into an unified form results. The function of joint max pooling method is to impose a consistency of distribution between objectness score and shape depiction, while reduce training difficulty by only focusing attention on part of the image. Finally parameterized shape description will be transformed into a general segmentation results by our shape generation layer. The whole network can be jointly trained end-to-end, and intermediate results (parameterized shape depiction) can be directly extracted as morphological statistics of object shape. Furthermore, our scnet can be easily converted to solve the object detection task, such as scene text detection, by regarding the detection bound box as a rectangular object.

2. Joint Region Proposal Netwrok

2.1. Overview

The complete pipeline of proposed method is illustrated in Figure 1. They are jointly trained end-to-end and consists of three components.

Analogy to famous Region Proposal Network [10], the first component takes an image as input and produces a coarse objectness score map and associated shape parameters map. Different to original RPN, the rectangular proposals are replaced by outline shape parameters of the object that it belongs to, and a dense prediction strategy is adopted which predicts an object for each position of input image. The objectness score is the predicted probability of the position being inside an object and shape depiction is a set of parameters, which depict outline shape of possible object it belonging to. Our model first find out all the possible object from predicted score map, and then depict its predicted shape by the shape parameter map. For instance in vesicle segmentation, the object shape depiction can be formulated by $[\theta, x_c, y_c, a, b]$, including the angle of major axis, coordinate of center point and two elliptical axis length of the object as in Figure 1, which restricts the segmented object to be ellipse. Especially, if only the shape constraint can be formulated by parameterization, our scnet can integrate this constraint into the network. The architecture of the first component is based on the publicly available DeepLab-LargeFov model [4], which modifies VGG-16 net [13] to be FCN [9] and introduces zeros into the filters to enlarge its Field-Of-View. At the end of DeepLab, another regression layer is introduced to predict shape parameter.

We develop a joint max pooling operation as our second component, which fuses the results from first components into a raw segmentation map. Especially, the existing region-based detectors lack intrinsic consistency, of which the results were dividedly optimized. However our joint max pooling can impose a consistency on the preceding outputs, restricting the highest score corresponding to the most accurate proposal. Furthermore, the implementation of joint max pooling is very efficient, since the residual error is sparse and can be correctly propagated. The whole joint max pooling operation can be easily expanded to other pooling operations.

The third component in our framework is a novel object generation layer, which transforms the raw segmentation map to the final segmentation prediction. The shape parameters map can be learnt directly from the segmentation

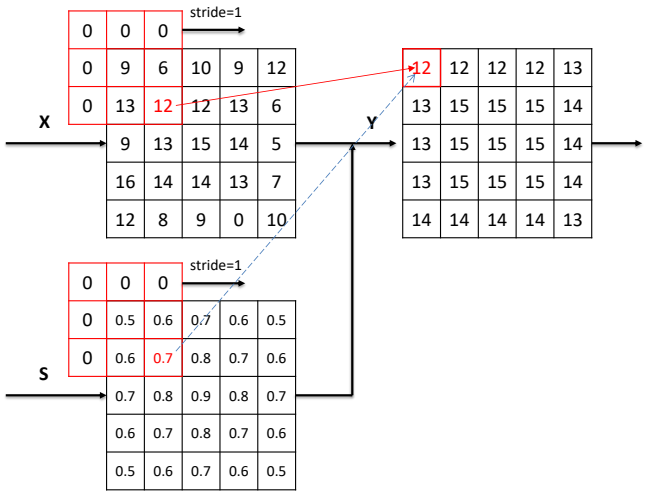

Figure 2. A diagram of joint max pooling process in Eq .4. Inputting two matrices $\mathbf{X}$ and $\mathbf{S}$, a 3×3 max pooling window is applied to $\mathbf{S}$ with padding operation for the top left position. Recording the position of the max valve 0.7 in $\mathbf{S}$, we output the value 12 in corresponding position in $\mathbf{X}$. The process is repeated with pooling window sliding.

results without preparing any other ground truth. Therefore, the accurate statistical shape parameters of each object can be very conveniently obtained.

We first introduce our joint pooling operation in Sec. 3.2, and then we extend it to a trainable system in Sec. 3.4. Finally the object generation is applied to generate the final object segmentation result.

2.2. Joint max pooling

Original max pooling operation takes a 2-D signal $\mathbf{X}$ as input and outputs a filtered signal $\mathbf{Y}$:

$$y_{\mu,\nu} = max(\{x_{i,j}|x_{i,j} \in \mathbf{P}\}) \tag{1}$$

where $\mathbf{P}$ is a local field of $\mathbf{X}$ associated to $y_{\mu,\nu}$ in $\mathbf{Y}$: Intuitively, information carried by the strongest x in $\mathbf{X}$ can be propagated to next layer, while the other x are abandoned. And the max pooling process can be further expressed as:

$$y_{\mu,\nu} = \sum_{i,j} x_{i,j} b_{i,j} \qquad x_{i,j} \in \mathbf{P}, b_{i,j} \in \mathbf{B} \tag{2}$$

where $\mathbf{B}$ is a binary matrix, whose elements are all zero except for the position where x is maximum in $\mathbf{P}$. In this formulation, $\mathbf{B}$ acts like an "indictor" determining which x can be propagated to next layer. And in Eq 2, the criterion of indictor is who is bigger. Especially most existing pooling methods can be interpreted by an "indictor" with different judgment criteria. Based on this intuition, a natural question arises that whether the criterion can be learnt in terms of different task. That is, the indictor is no longer fixed before task beginning and can be learnt during training stage, in which way, the pooling operation will be more powerful and flexible.

Followed by this discussion, the joint max pooling method is proposed, as illustrated in Figure 2. The passing x is no longer determined by its value, instead we learn a more intelligent "indictor" to determine which x is more valuable. The whole process can be regarded as a split version of pooling, whose input is split into two parts. In practice, $\mathbf{B}$ is hard to be directly learnt as binary, instead we learn a score matrix $\mathbf{S}$ that evaluates the importance of information carried by x. Therefore only the information with highest importance can be propagated:

$$\overline{s} = max(\{s_{i,j}|s_{i,j} \in \mathbf{S}\} \tag{3}$$

$$y_{\mu,\nu} = \sum_{i,j} x_{i,j} g(s_{i,j} - \overline{s}) \quad x_{i,j} \in \mathbf{P} \tag{4}$$

$$g(t) = \begin{cases} 1 & if\ t >= 0 \\ 0 & else \end{cases} \tag{5}$$

The intrinsic consistency of join max pooling, imposing on its two inputs, is very suitable to associate shape depiction and object location together, as only the shape predictions with highest score can be propagated and updated. As shown in Fig, $\mathbf{X}$ refers to the predicted shape parameters map, while $\mathbf{S}$ is the objectness score map. Each channel of shape parameters map $\mathbf{X}$ shares the $\mathbf{S}$. Explicitly, we use kernel size 7×7 with stride 1 and iterate several times to let the shape prediction with high score cover a broader area. And the output of joint pooling is a new shape prediction map, of which those predictions with low score have been all filtered out.

Moreover in a natural image, most of the area is background containing no object, and predicting shape in these regions is a wasting of energy. Eq. 2 still output those predictions with local max score, although they are such low scores that we believe it's background. Therefore Eq. 4 is further improved by:

$$y_{\mu,\nu} = \begin{cases} \sum_{i,j} x_{i,j} g(s_{i,j} - \overline{s}) & if\ \overline{s} \geq \tau \\ 0 & else \end{cases} \tag{6}$$

where τ is the min threshold of score that is believed to be inside the objective region. If $\overline{s}$ is lower than threshold τ, all the $y_{\mu,\nu}$ associated to $\mathbf{P}$ will be set to 0, which means that all the $x_{i,j} \in \mathbf{P}$ belongs to background.

Our joint max pooling operation is effective. On the one hand, the joint pooling method reduces the difficulty of learning shape predictions, since only the positions

with highest objectness score are required to predict accurate shape parameters. Besides, the information of the region with high objectness score is more abundant. On the other hand, as all the residual errors have been converged into the positions with high score, the shape prediction in these positions will be more accurate, which will be illustrated in 3.3.

2.3. Trainable joint pooling operation

One important contribution of our joint pooling is that the residual error can be correctly back propagated to its two inputs. This makes the joint pooling operation be a trainable layer in any network architecture and our scnet become a fully trainable system.

The forward pass of Eq. 6 is illustrated in Figure 2 and next we will demonstrate the back propagation of joint max pooling backpropagation in a general form. Exactly, the objective of back propagation should be: (i) improving the objetness scores of the object area; (ii) minimizing the error of predicting shape parameters of positions with local highest objectness score. (iii) making the local maximum points in objectness score map be sparse and moving them to the center of object as far as possible.

The third objective is to further reduce the calculation and difficulty of learning for shape predicting. Specially in Fig, each objectness score map is assumed to not only influence the output but also feeds a subsequent layers, thus also receiving gradient contributions $\frac{\partial L}{\partial s_{i,j}}$ from the next layer during back-propagation. Defining $\mathbf{U}$ as the $\{y_{\mu.\nu}\}$ set related to $x_{i,j}$ and m as the size of $\mathbf{U}$, the back propagation can be expressed by:

$$\frac{\partial L}{\partial x_{i,j}} = \begin{cases} \frac{1}{m} \sum\limits_{y_{\mu,\nu} \in \mathbf{U}} \frac{\partial L}{\partial y_{\mu,\nu}} & if\ s_{i,j} \geq max(\overline{s}, \tau) \\ 0 & else \end{cases} \tag{7}$$

if $s_{i,j} \geq max(\overline{s}, \tau)$:

$$\begin{aligned} \frac{\partial L}{\partial s_{i,j}} &= \frac{\partial L}{\partial s_{i,j}} + \frac{1}{m} \sum_{y_{\mu,\nu} \in \mathbf{U}} \frac{\partial L}{\partial y_{\mu,\nu}} \frac{\partial y_{\mu,\nu}}{\partial s_{i,j}} \\ &= \frac{\partial L}{\partial s_{i,j}} + \frac{1}{m} \sum_{y_{\mu,\nu} \in \mathbf{U}} \frac{\partial L}{\partial y_{\mu,\nu}} x_{i,j} \frac{\partial g}{\partial s_{i,j}} \\ &\approx \frac{\partial L}{\partial s_{i,j}} + \lambda sign(\frac{1}{m} \sum_{y_{\mu,\nu} \in \mathbf{U}} \frac{\partial L}{\partial y_{\mu,\nu}} x_{i,j}) \end{aligned} \tag{8}$$

else:

$$\begin{aligned} \frac{\partial L}{\partial s_{i,j}} &= \frac{\partial L}{\partial s_{i,j}} + \frac{1}{m} \sum_{y_{\mu,\nu} \in \mathbf{U}} \frac{\partial L}{\partial y_{\mu,\nu}} \frac{\partial y_{\mu,\nu}}{\partial s_{i,j}} \\ &\approx \frac{\partial L}{\partial s_{i,j}} - \lambda g(\frac{1}{m} \sum_{y_{\mu,\nu} \in \mathbf{U}} || \frac{\partial L}{\partial y_{\mu,\nu}} ||) \end{aligned} \tag{9}$$

where $sign$ is the general signal function and λ control the change amplitude of $s_{i,j}$, because $\mathbf{Y}$ is very sensitive to $\mathbf{S}$.

Eq. 7 modifies the standard back propagation of max-pooling by changing the residual convergence from the position of max x to max p, which emphasize improving the accuracy of $x_{i,j}$ with local highest $s_{i,j}$. When $s_{i,j} \geq max(\overline{s}, \tau)$, Eq. 8 follows the standard chain rules to infer the gradients of $s_{i,j}$. In order to avoid gradient vanishing caused by $\frac{\partial g}{\partial s_{i,j}}$, we let $\frac{\partial g}{\partial s_{i,j}} = 1$ when $s_{i,j} \geq 0$. Furthermore, as $-\frac{\partial L}{\partial y_{\mu,\nu}} x_{i,j}$ reflects whether $x_{i,j}$ will move towards $\mathbf{0}$, Eq. 8 will increase s of positions with negative $\frac{\partial L}{\partial y_{\mu,\nu}} x_{i,j}$ which means i, j is probably inside an object. Since only the local maximum s can be updated in Eq. 9, the final local maximum points are sparse. When $s_{i,j} < max(\overline{s}, \tau)$, the second term in Eq. 9 indicates that there exist prediction errors here, although $\overline{s}$ believe no object is here. Therefore we design to enlarge $s_{i,j}$ by λ to improve the objectness score in this position, where exists missing detection.

2.4. Shape Generation Layer

The most difficult dilemmas of dense segmentation tasks are the touching and rough boundary problems, as shown in. However as our scnet specially learn the shape information of each object in the shape parameters map, these problems can well be handled using the final shape generation layer.

For examples, if we depict each vesicle shape by a set of parameters $x = [\theta, x_c, y_c, a, b]$, respectively representing the angle of major axis, coordinate of center point and two elliptical axis length, the shape generation layer will output a result o indicating the corresponding label of the position according to x:

$$\begin{aligned} \xi_1 &= cos(\theta)(i - x_c) + sin(\theta) * (j - y_c) \\ \xi_2 &= -sin(\theta)(i - x_c) + cos(\theta) * (j - y_c) \\ o_{i,j} &= (sign(1 - \frac{\xi_1^2}{a^2} - \frac{\xi_2^2}{b^2}) + 1)/2 \end{aligned} \tag{10}$$

The back propagation follows the standard back-propagation derivation. The gradient is then further propagated onto the preceding joint max pooling, and the whole network will be jointly optimized.

References

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision*, pages 524–540. Springer, 2016. 1

[2] H. Chen, X. Qi, L. Yu, and P.-A. Heng. Dcan: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016. 1

[3] 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*, pages 4545–4554, 2016. 1

[4] 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. 1, 2

[5] R. Fernández-Busnadiego, S. Asano, A.-M. Oprisoreanu, E. Sakata, M. Doengi, Z. Kochovski, M. Zürner, V. Stein, S. Schoch, W. Baumeister, et al. Cryo–electron tomography reveals a critical role of rim1α in synaptic vesicle tethering. *J Cell Biol*, 201(5):725–740, 2013. 1

[6] R. Fernández-Busnadiego, B. Zuber, U. E. Maurer, M. Cyrklaff, W. Baumeister, and V. Lučić. Quantitative analysis of the native presynaptic cytomatrix by cryoelectron tomography. *The Journal of cell biology*, 188(1):145–156, 2010. 1

[7] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis. Convexity shape prior for segmentation. In *European Conference on Computer Vision*, pages 675–690. Springer, 2014. 1

[8] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3129–3136. IEEE, 2010. 1

[9] 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*, pages 3431–3440, 2015. 2

[10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[11] 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*, pages 234–241. Springer, 2015. 1

[12] L. A. Royer, D. L. Richmond, C. Rother, B. Andres, and D. Kainmueller. Convexity shape constraints for image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 402–410, 2016. 1

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[14] K. Sirinukunwattana, D. R. Snead, and N. M. Rajpoot. A stochastic polygons model for glandular structures in colon histology images. *IEEE transactions on medical imaging*, 34(11):2366–2378, 2015. 1

[15] E. Strekalovskiy and D. Cremers. Generalized ordering constraints for multilabel optimization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2619–2626. IEEE, 2011. 1

[16] O. Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision*, pages 454–467. Springer, 2008. 1

[17] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 1

[18] A. A. S. J. S. Zheng and P. Torr. Higher order conditional random fields in deep neural networks. 1