2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)
April 4-7, 2018, Washington, D.C., USA

PIXEL-WISE NEURAL CELL INSTANCE SEGMENTATION

Jingru Yi[1], Pengxiang Wu[1], Daniel J. Hoeppner[2], Dimitris Metaxas[1]

[1] Department of Computer Science, Rutgers University, NJ, USA
[2] Astellas Research Institute of America, IL, USA

ABSTRACT

Accurate cell instance segmentation plays an important role in the study of neural cell interactions, which are critical for understanding the development of brain. These interactions are performed through the filopodia and lamellipodia of neural cells, which are extremely tiny structures and as a result render most existing instance segmentation methods powerless to precisely capture them. To solve this issue, in this paper we present a novel hierarchical neural network comprising object detection and segmentation modules. Compared to previous work, our model is able to efficiently share and make full use of the information at different levels between the two modules. Our method is simple yet powerful, and experimental results show that it captures the contours of neural cells, especially the filopodia and lamellipodia, with high accuracy, and outperforms recent state of the art by a large margin.

***Index Terms*—** Neural cell, instance segmentation, deep learning, cell segmentation, transpose convolution

1. INTRODUCTION

The cellular mechanisms engaged during the standard specification of neurons, astrocytes, and oligodendrocytes from a single neural stem cell are among the vital mysteries in neural science [1]. In the lineage history, the neural cells contact each other frequently through their filopodia and lamellipodia, while undergoing mitosis, movement, and morphology changes. Thus, accurate capture of the cell interactions is critical to the understanding of neural cell behavior and normal brain development [1]. With the help of real-time imaging system, it is highly possible to study such interactions through vision techniques, such as segmentation, tracking and detection [2], among which accurate segmentation is crucial for pinpointing the time when cells communicate.

Recent years have witnessed the significant improvement of semantic image segmentation due to deep neural networks [3, 4, 5]. In [3], Long *et al*. introduced the groundbreaking fully convolutional networks (FCN) which greatly advanced the accuracy of semantic segmentation. However, the deconvolution operation of FCN simply employs fixed bilinear interpolation, making it difficult to obtain accurate boundaries for highly non-linear objects [5]. To mitigate the limitations of FCN, Noh *et al.* [5] proposed to learn a deep deconvolution network featured with learnable deconvolution and unpooling layers, which are able to generate dense and precise object segmentation masks. In a similar fashion, Ronneberger *et al.* [4] combined the features on the lower layers with the upsampled features on the higher layers, and thereby increased the segmentation precision a lot.

However, semantic segmentation is unaware of the individual object instance [6]. To understand the interactions between neural cells, identifying the cell instances so as to separate them is of great importance. Therefore, the goal of this paper is to develop a framework that enables accurate instance segmentation for neural cells.

Instance segmentation is a challenging task as it requires precise detection of objects along with correct segmentation masks. In [6], Dai *et al*. proposed a multi-task network cascades (MNC) approach that segments the objects from the bounding-box proposals, followed by object classification. However, the feature warping and resizing in the ROI pooling step make it lose the important spatial details, thus hurting the segmentation performance. In [7], Li *et al.* further developed the idea of MNC and presented a fully convolutional instance-aware semantic segmentation (FCIS) framework that detects and segments the object instances jointly. However, FCIS tends to create spurious edges and exhibit systematic artifacts on overlapping objects, and is unable to accurately delineate the contours of objects with fine boundary details, which are just the case for neural cells (see Fig. 2).

In this paper, we propose a novel pixel-wise instance segmentation method that jointly performs object detection and segmentation. However, different from previous approaches, our model does not fix the ROI size and thus gets rid of the ROI misalignment issue caused by warping, as is the case for [6]. Furthermore, unlike [6] and [7], to achieve pixel-accurate segmentation, the feature maps of upstream convolutional layers are fully shared for detection and segmentation subtasks in a fashion similar to [4], but without suffering from the cropping issues. Our object detection is based on a light-weight SSD detector [2, 8], which is simple, fast and powerful. Experimental results demonstrate that our method outperforms MNC and FCIS by a large margin for the neural cell instance segmentation task.

978-1-5386-3636-7/18/$31.00 ©2018 IEEE

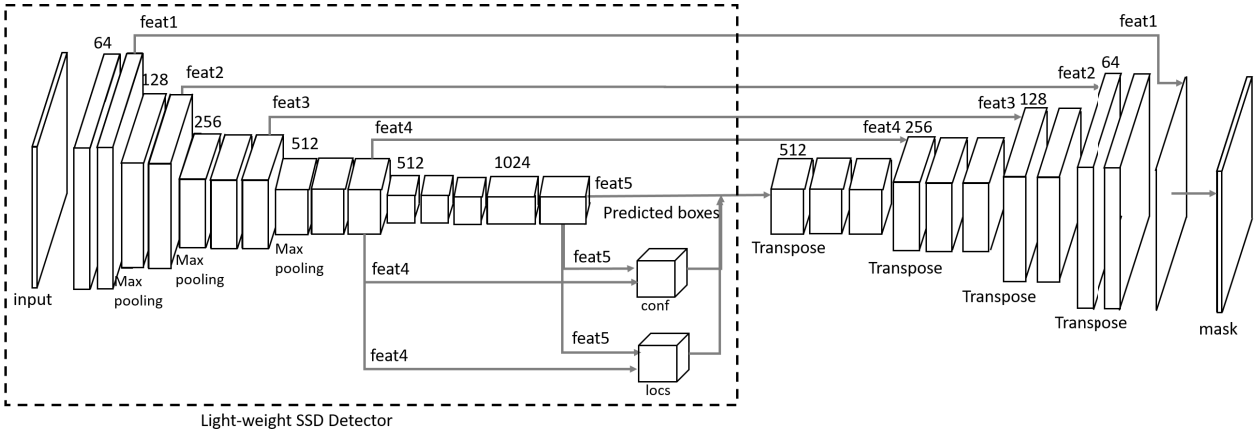


Fig. 1. Schema of our pixel-wise instance segmentation framework. The architecture consists of two parts: object detection and segmentation, of which the object detector streams down confidence and bounding box offsets to the higher layers for object mask prediction. The feature maps of the upstream layers are shared for both the detection and segmentation subtasks.

2. METHODS

Our pixel-wise instance segmentation network (as shown in Fig. 1) comprises two parts: object detection and segmentation, which are trained jointly. The outputs of object detector are confidence scores and bounding box offsets, based on which object instance masks are predicted.

2.1. Light-weight SSD detector

The upstream part of our network is a light-weight SSD detector [2, 8], which eliminates the widely-used proposal generation and subsequent feature resampling [6, 9, 10]. In this way, the detector encapsulates all the computation in a single network, making it easier to train and more flexible to apply to instance segmentation.

As shown in Figure 1, the base model of the light-weight SSD detector [2] is the VGG-16 [11] deep convolutional networks. In the forward pass, the size of feature maps shrinks gradually, thereby capturing object features at different scales. Considering that the cell shapes vary significantly across different neural cells, we choose to utilize feature maps at different levels to better locate cells. In particular, two feature maps (feat4 and feat5) are selected and combined to handle cells of various sizes.

Default boxes. Instead of generating proposals, the SSD detector discretizes the feature maps (feat4 and feat5) and generates fixed-size default boxes with different aspect ratios and scales. In obtaining the default boxes, an $m \times m$ feature map is divided into 1×1 cells, and the normalized cell centers are set as the centers of the default boxes, i.e., $(cx_d, cy_d) = (\frac{i+0.5}{m}, \frac{j+0.5}{m}), i, j = 0, 1, \cdots, m-1$. By this design, default boxes are quite powerful to match objects of various sizes at different locations. In practice, the aspect ratios are $a_r \in \{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, 2, 3, 4\}$ for both feat4 and feat5, and the scales are set to be $s_{\text{feat4}} \in \{0.04, 0.07, \sqrt{0.07^2 + 0.15^2}\}$ for feat4 and $s_{\text{feat5}} \in \{0.15, 0.29, \sqrt{0.29^2 + 0.33^2}\}$ for feat5. Then the width and height of default boxes are calculated as $w = s\sqrt{a_r}, h = s/\sqrt{a_r}$, respectively.

Encoding ground-truth boxes. After obtaining the default boxes, we encode the object localization information into the default boxes and generate the encoded ground truth.

The encoding steps are as follows. First, we match each default box to all the ground truth boxes. If the Jaccard value is higher than 0.5, we set the label of this default box as 1, otherwise 0. Then the offset between this default box and the best-matched ground truth box is saved as the offset box. The ground truth vector is then encoded with the offset box and the box label. The offset box $g = (cx, cy, w, h)$ is computed as follows [8]:

$$cx = (cx_g - cx_d)/w_d \quad (1)$$

$$cy = (cy_g - cy_d)/h_d \quad (2)$$

$$w = \log(w_g/w_d) \quad (3)$$

$$h = \log(h_g/h_d), \quad (4)$$

where (cx, cy) is the center of the encoded offset box, and The subscript index g and d refer to the encoded offset box and the default box, respectively.

ROI prediction. After training with encoded ground truth vectors, in the forward inference SSD detector predicts the offset boxes for each object. These predicted offset boxes are then decoded to obtain the real bounding boxes of objects.

2.2. Pixel-wise segmentation

In the detection process, max pooling operation is utilized to sequentially abstract the neuron activations with fewer rep-

resentative value. In this way, the higher layers only retain robust activations, which are helpful for detection and classification [5]. However, one deficiency associated with higher layers is that they lose the spatial details, which play a significantly important role in pixel-accurate segmentation.

To remedy the loss of spatial details, we unpool the feature maps and employ transposed convolution arithmetic [12] to transform them to have the same size with the input image, as shown in Fig. 1. Besides, the feature maps of SSD detector at different levels are propagated to the transposed convolutional layers, and in this way the contexts of objects at different scales could be fully reused to predict the lost spatial information. Note that this layer linking strategy is similar to [4], but does not suffer from the cropping issues since the contracting and unpooling parts are exactly symmetric.

For each generated ROI, we predict the object mask within it, using the features bounded by the counterparts of the given ROI at each layer. However, since the max pooling operation used in our network always contracts the feature map twice, there would be misalignment between the floating-number ROIs from different layers. For instance, given ROI of size 15×15, then after max pooling it reduces to 7.5×7.5, which can be naturally rounded up to 8×8; however, in the unpooling process this 8×8 ROI will be scaled up twice to 16×16, which is inconsistent with the original ROI. To solve this issue, for the unpooled 16×16 ROI, we just simply remove its last row and column corresponding to the extra 0.5×0.5 floating-point parts due to rounding up to convert the ROI to 15×15. With this simple strategy, our network is able to precisely align the predicted instance mask with the input image, as demonstrated in Fig. 2.

2.3. Loss function

The loss function for network training is composed of three parts: confidence scores, object locations, and segmentation masks

$$L = \frac{1}{N_{\text{pos}}}(L_{\text{conf}} + \alpha L_{\text{locs}}) + L_{\text{masks}}, \tag{5}$$

where α is the weight. The object location offset loss is defined as [8, 13]:

$$L_{\text{locs}} = \sum_{i \in \text{pos}} \sum_{m \in \{cx,cy,w,h\}} \text{smooth}_{L_1}(l_i^m - g_i^m), \tag{6}$$

$$\text{smooth}_{L_1}(z) = \begin{cases} 0.5z^2 & \text{if } |z| < 1 \\ |z| - 0.5 & \text{otherwise} \end{cases}, \tag{7}$$

where $i \in$ pos denotes the set of positive predicted boxes (whose encoded labels are positive), and l_i^m and g_i^m refer to the predicted and encoded offset boxes, respectively. The confidence score loss is calculated as the binary cross-entropy:

$$L_{\text{conf}} = -\sum_i (x_i \log p_i + (1 - x_i) \log(1 - p_i)) \tag{8}$$

where x_i is the encoded ground truth label, and p_i is the predicted confidence score. The segmentation mask loss is also modeled as a binary cross-entropy:

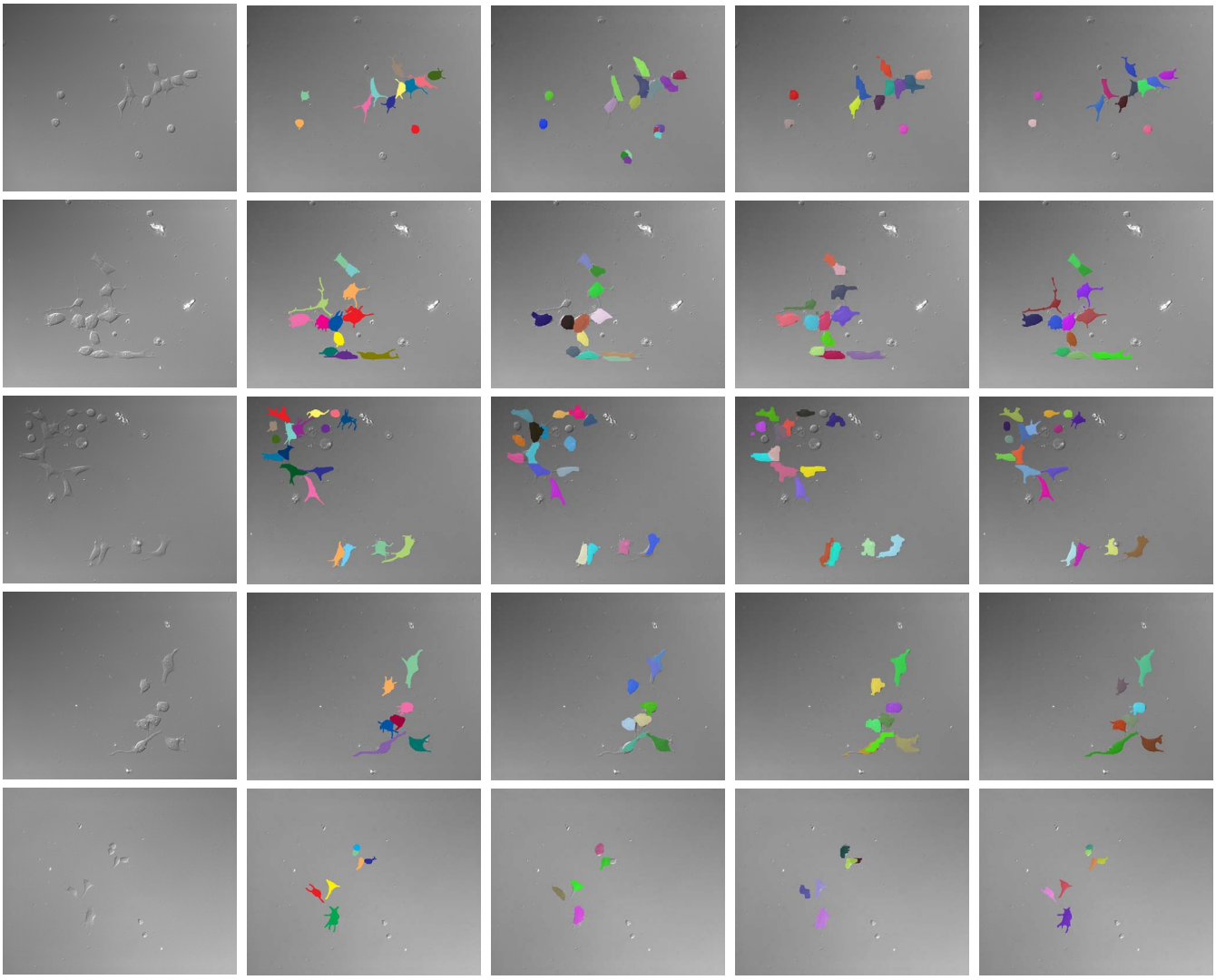
$$L_{\text{masks}} = -\frac{1}{N}\sum_j^N \sum_i (t_{ij} \log p_{ij} + (1 - t_{ij}) \log(1 - p_{ij})) \tag{9}$$

where p_{ij} and t_{ij} are respectively the predicted and ground truth mask values at position i for the jth positive predicted bounding box (whose overlap with the ground truth box exceeds a certain threshold), and N is the total number of positive predicted bounding boxes.

3. EXPERIMENTS AND RESULTS

The neural cell images used in our experiment came from a series of time-lapse microscopy videos, from which we sampled 386 images for training, 129 for validation, and 129 for testing. The size of the cell images is 640×512, and the ground truths of the cell instance masks were labeled by experts. As for training, the parameters of the contracting part of the network were fine-tuned with the VGG-16 pretrained weights on ImageNet [14], while the remaining part was initialized with random weights sampled from standard Gaussian distribution. We flipped the images horizontally and vertically to augment the training set. The network was implemented with Pytorch [15], and was trained and tested on a single Nvidia K40 GPU. The average inference time of the trained model is 0.6s per image.

The predicted object masks will be evaluated by AP (average precision [16]) at mask-level IoU (intersection-over-union) thresholds of 0.5 and 0.7. When the mask-level IoU between a single predicted object mask and the ground-truth mask is greater than 0.5 or 0.7, the detection of the corresponding object will be considered as true positive, otherwise considered as false positive. In the former case, the ground-truth object will be recorded as "detected", which means the other predicted results for this same object will be considered as false positive. Finally, with the above statistics we compute the precision/recall curve, and utilize AP [16] metric to summarize the shape of this curve, giving an evaluation which measures both instance detection and segmentation accuracy, as with [6] and [7]. Besides, we also measure the average mask IoU at thresholds 0.5 and 0.7. The results are reported in Table 1, which shows that our method outperforms MNC and FCIS by a large margin, especially for AP at mask-level IoU 0.7. We also demonstrate the segmentation results qualitatively in Fig. 2, and observe that both MNC and FCIS failed



(a) Neural cell image (b) Ground truth (c) MNC [6] (d) FCIS [7] (e) Ours

Fig. 2. Instance segmentation results of MNC [6], FCIS [7] and our model, where the cell instances are denoted by different colors. Compared to MNC and FCIS, our method is more accurate and is able to capture the tiny structures, particularly the filopodia and lamellipodia, of neural cells.

to capture fine boundary details (particularly the filopodia and lamellipodia), while our model was able to delineate them precisely. As our goal is to study the interactions of the neural cells, accurate segmentation of filopodia and lamellipodia is extremely important, indicating the potential value of our method to neuroscience research.

Model	AP@0.5	AP@0.7	IoU@0.5	IoU@0.7
MNC [6]	48.72	11.37	62.71	75.47
FCIS [7]	66.02	7.13	64.85	75.07
Ours	**85.7**	**70.94**	**78.84**	**81.22**

Table 1. Quantitative comparisons of MNC [6], FCIS [7] and our method. Amongst them, our method achieves the best results and outperforms the other two by a large margin.

4. CONCLUSION

In this paper, we present a novel method for pixel-wise instance segmentation of neural cells. Compared to recent state of the arts, our method achieves better accuracy and is able to capture the tiny boundary structures, particularly the filopodia and lamellipodia, of neural cells. This characteristic reveals that our method is of great potential value to the study of neural cells and the neuroscience research.

5. REFERENCES

[1] Rea Ravin, Daniel J. Hoeppner, David M. Munno, Liran Carmel, Jim Sullivan, David L. Levitt, Jennifer L. Miller, Christopher Athaide, David M. Panchision, and Ronald D.G. McKay, "Potency and fate specification in cns stem cell populations in vitro," *Cell Stem Cell*, vol. 3, no. 6, pp. 670 – 680, 2008.

[2] Jingru Yi, Pengxiang Wu, Daniel J Hoeppner, and Dimitris Metaxas, "Fast neural cell detection using lightweight ssd neural network," in *Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 860–864.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[5] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, "Learning deconvolution network for semantic segmentation," *CoRR*, vol. abs/1505.04366, 2015.

[6] Jifeng Dai, Kaiming He, and Jian Sun, "Instance-aware semantic segmentation via multi-task network cascades," *CoRR*, vol. abs/1512.04412, 2015.

[7] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei, "Fully convolutional instance-aware semantic segmentation," *CoRR*, vol. abs/1611.07709, 2016.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017.

[10] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[11] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[12] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *ArXiv e-prints*, Mar. 2016.

[13] Ross B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[15] "Pytorch," https://github.com/pytorch/pytorch, accessed: 10-08-2017.

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.