Weakly-Supervised Nucleus Segmentation Based on Point Annotations: A Coarse-to-Fine Self-Stimulated Learning Strategy

Kuan Tian[1], Jun Zhang[1], Haocheng Shen[1], Kezhou Yan[1], Pei Dong[1], Jianhua Yao[1], Shannon Che[2], Pifu Luo[2(✉)], and Xiao Han[1(✉)]

[1] Tencent AI Lab, Shenzhen, Guangdong, China
haroldhan@tencent.com
[2] KingMed Diagnostics Co., Ltd., Guangzhou, Guangdong, China
Siweipathology@qq.com

Abstract. Nucleus segmentation is a fundamental task in digital pathology analysis. However, it is labor-expensive and time-consuming to manually annotate the pixel-level full nucleus masks, while it is easier to make point annotations. In this paper, we propose a coarse-to-fine weakly-supervised framework to train the segmentation model from only point annotations to reduce the labor cost of generating pixel-level masks. Our coarse-to-fine strategy can improve segmentation performance progressively in a self-stimulated learning manner. Specifically, to generate coarse segmentation masks, we employ a self-supervision strategy using clustering to perform the binary classification. To avoid trivial solutions, our model is sparsely supervised by annotated positive points and geometric-constrained negative boundaries, via point-to-region spatial expansion and Voronoi partition, respectively. Then, to generate fine segmentation masks, the prior knowledge of edges in the unadorned image is additionally utilized by our proposed contour-sensitive constraint to further tune the nucleus contours. Experimental results on two public datasets show that our model trained with weakly-supervised data (i.e., point annotations) achieves competitive performance compared with the model trained with fully supervised data (i.e., full nucleus masks). The code is made publicly available at https://github.com/tiankuan93/C2FNet.

Keywords: Nucleus segmentation · Weakly-supervised learning · Point-based supervision

1 Introduction

The recent success of deep learning approaches for image segmentation in natural image analysis is generally supported by large-scale fully annotated datasets.

K. Tian and J. Zhang—These authors contribute equally to this paper.

A. L. Martel et al. (Eds.): MICCAI 2020, LNCS 12265, pp. 299–308, 2020.
https://doi.org/10.1007/978-3-030-59722-1_29

Although several deep-learning-based nucleus segmentation methods have been proposed [8,9,11,12], it is still challenging to segment nuclei from pathological images, due to limited training data with full nucleus masks. Generally, it is labor-expensive and time-consuming to perform the full mask annotation. Alternatively, it is much easier to annotate the nuclei with points.

Currently, there are a few studies that focus on the problem of segmenting nuclei with point supervision. To train nucleus segmentation model with only point annotations, extra supervised information, including geometric diagram and clustering labels have been employed [2–4]. For example, Qu et al. [3] proposed a weakly-supervised method for nucleus segmentation based on point annotation in H&E histopathology images, which extracts pixel-level labels by using the Voronoi diagram and k-means clustering algorithm. Then, Chamanzar et al. [4] further modified this method to detect and segment nuclei in immunohistochemistry (IHC) images by using local pixel clustering in every Voronoi sub-region and repel encoding. However, these methods do not pay attention to the nucleus boundary. Differently, Nishimura et al. [2] proposed to a postprocessing method to segment the individual nucleus with graph-cut after obtaining the nucleus region map. Generally, it is difficult to rectify the large bias by independent post-processing. Therefore, Yoo et al. [5] extended a blob generation method (training with point supervision) [1] for nucleus segmentation with an auxiliary network, in which an auxiliary network helps the segmentation network to recognize nucleus boundaries. For the same purpose, Qu et al. [3] employed a dense CRF loss for model refinement in nucleus segmentation.

Accordingly, we would like to develop a method that can integrate the benefits of using pixel clustering and boundary attention. In this paper, we propose a coarse-to-fine framework that can improve the segmentation performance progressively in a self-stimulated learning manner. Specifically, to generate coarse segmentation masks, we employ a self-supervision strategy using clustering to perform the binary classification. To avoid trivial solutions, our model is sparsely supervised by annotated positive points and geometric-constrained negative boundaries, via point-to-region spatial expansion and Voronoi partition. Then, to generate fine segmentation masks, the prior knowledge of edges in the unadorned image is additionally utilized by our proposed contour-sensitive constraint to further tune the nucleus contours. By doing so, both coarse information (i.e., the roughly mask generated by stimulated learning from point annotation) and contour information (i.e., the contour obtained by unadorned image) can be progressively integrated into the learning model in the whole framework, by utilizing our rectified supervisions. Experiments show that our model trained with weakly-supervised data achieves competitive performance compared with the model trained with fully supervised data on MoNuSeg and TNBC datasets.

2 Method

As shown in Fig. 1, our method has two major stages for training the fully convolutional networks (FCN). The first stage obtains the initial coarse nucleus

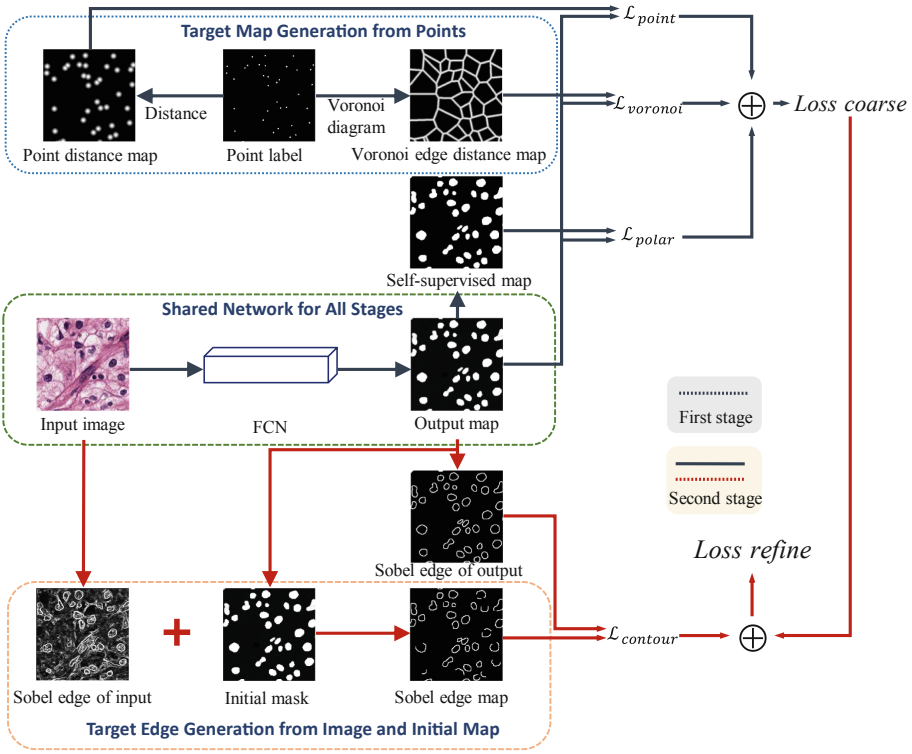


Fig. 1. Framework of our proposed method.

masks for all training data with self-supervised learning and estimated *distance* maps. The second stage further refines the FCN with an additional contour constraint. In the application stage, our FCN model can directly perform the inference with the well trained FCN.

2.1 Coarse Segmentation Estimation

Our target is to generate coarse segmentation masks in the first training stage. Intuitively, we can perform binary classification with clustering via self-supervised learning (i.e., deep clustering [6]). However, typical clustering has the problem of trivial solutions. An optimal decision boundary is to assign all pixels to a single class. While point annotations provide us necessary positive pixels that are too sparse and one class only. Therefore, we would like to transform the point annotations to more informative supervision maps in the first place. With the generated supervision maps, we can train an FCN model end-to-end to obtain the coarse segmentation masks.

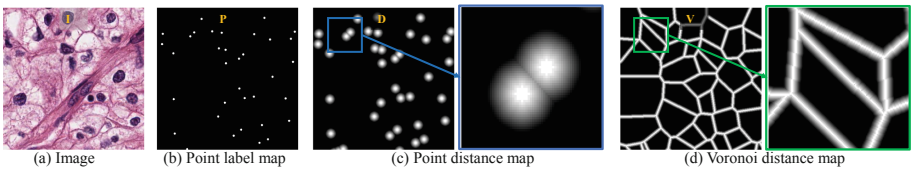


Fig. 2. Sparse supervision maps for segmentation.

Maps for Supervision. We denote the image as $\mathbf{I}$ and the positive point annotation map as $\mathbf{P}$. We intend to generate two distance maps that focus on reliable positive and negative pixels, respectively.

1) We propose a *point distance map* (i.e., $\mathbf{D}$) focusing on positive pixels with high confidence. We assume that the annotated point for each nucleus is near the center of the nucleus. Then, we perform a distance filter to point annotations to dilate the dot to a local region with decreasing response, which is considered reliable nucleus supervision, as shown in Fig. 2(c). Mathematically, each element $d_{i,j}$ (i and j are the coordinates in the image space) in $\mathbf{D}$ is calculated as

$$d_{i,j} = max(0, 1 - \alpha\sqrt{(i-m)^2 + (j-n)^2})) \tag{1}$$

where m and n are the coordinates of the nearest positive point in the positive point annotation map $\mathbf{P}$, and α is a scaling parameter to control the scale of distribution. Note that, a Gaussian-like filter could also be employed in our application to obtain the point distance map.

2) We propose another *Voronoi edge distance map* (i.e., $\mathbf{V}$) focusing on negative pixels with high confidence. Since most nuclei are convex and have the shape of ellipse, the Voronoi diagram, according to a given set of points, is an ideal partition of a plane into blocks. Therefore, we employ the Voronoi diagram to obtain the partition edges that are further dilated with the rapidly decreasing response using a distance filter (Eq. 1). This Voronoi edge distance map is utilized to describe reliable negative pixels, as shown in Fig. 2(d).

First Stage Sparsely Supervised Learning. To perform the self-supervised learning, we employ the polarization loss to guide the update of the weights (i.e., $\mathbf{W}$) in the FCN (denoted as f). Denote the output segmentation map as $\mathbf{S}$, with the probability value from 0 to 1, the polarization loss is calculated as

$$\mathcal{L}_{polar}(\mathbf{W}) = \| \, (f(\mathbf{I}) - H(\mathbf{S\text{-}0.5})) \, \|_F^2, \tag{2}$$

where the H(Heaviside step function) operation rectified the output segmentation map to the binary mask to realize the self-supervised learning. Note that, we do not require the function H to be differentiable, since we employ this function for generating the pseudo segmentation mask.

Besides, we calculate two *sparse* losses, named $\mathcal{L}_{point}$ and $\mathcal{L}_{voronoi}$, to guide the update of $\mathbf{W}$. Since the two maps only focus on partial positive and negative

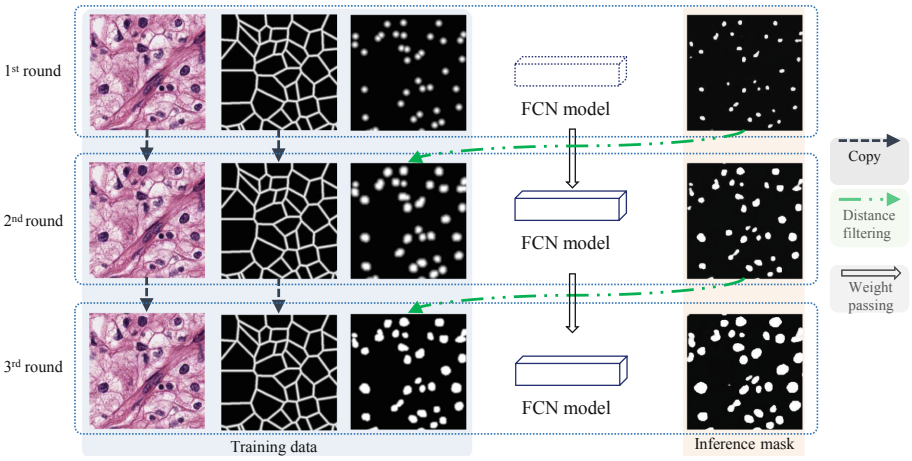


Fig. 3. Point-to-region coarse segmentation method.

pixels. The pixels without responses are the unknown pixels that should not be involved in calculating the loss. Therefore, the losses are sparsely calculated according to the following equations:

$$\mathcal{L}_{point}(\mathbf{W}) = \parallel ReLU(\mathbf{D}) \cdot (f(\mathbf{I}) - \mathbf{D}) \parallel_F^2, \tag{3}$$

$$\mathcal{L}_{voronoi}(\mathbf{W}) = \parallel ReLU(\mathbf{V}) \cdot (f(\mathbf{I}) - \mathbf{0}) \parallel_F^2, \tag{4}$$

where $\cdot$ is the pixel-wise product, and the ReLU operation here is to extract the reliable weight mask for sparse loss calculation. By doing this, $\mathcal{L}_{point}$ only focuses on the assured positive pixels, and $\mathcal{L}_{voronoi}$ only focuses on the assured negative pixels.

Generally, it is difficult to directly obtain satisfactory segmentation masks by training with such sparse constraints. While we could receive initial segmentation maps that are the expansion of our point annotations. Therefore, we iteratively train the segmentation model with the expanded point distance maps, which are updated by the latest trained model. The *point distance map* (i.e., $\mathbf{D}$) is updated according to Eq. 1, where the point annotation map $\mathbf{P}$ is replaced with the estimated segmentation mask (i.e., $\mathbf{S}_c$) from previous training round. The operation repeats two additional times to achieve reliable segmentation masks. As shown in Fig. 3, the silhouette of the nucleus gradually becomes clear by multiple training rounds. Note that, we employ the same Voronoi edge distance map for three iterations. Importantly, because the nuclei differ significantly in size in different images, it is a good idea to use the same size disk (up to the nucleus scale) as the nucleus area. Small nuclei will provide *wrong reliable* positive pixels. Therefore, we gradually fit the coarse segmentation that is more suitable for nuclei of different sizes.

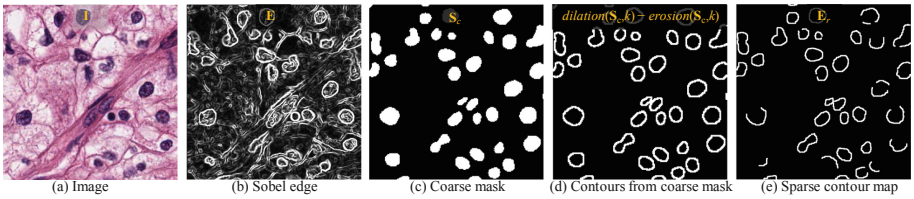
2.2 Contour Refinement

The contours of nuclei in coarse segmentation are not accurate. We propose to use an additional contour-sensitive constraint to refine the contours.

Contour Map for Supervision. For the observation that the colors of nucleus pixels are often different from the surrounding background pixels. We can extract the apparent contour (not necessary to be the complete contours for nuclei) of the input images as an additional supervision. Specifically, we first employ a Sobel operator to detect edges from the original images. Not surprisingly, there are lots of noisy edges in the edge map (i.e., $\mathbf{E}$), as shown in Fig. 4(b). Then, we refine the edge map by the coarse segmentation mask (i.e., $\mathbf{S}_c$), obtained in the *first stage*, to eliminate the unnecessary Sobel edges. The refined edge map (i.e., $\mathbf{E}_r$) is obtained as

$$\mathbf{E}_r = (dilation(\mathbf{S}_c, k) - erosion(\mathbf{S}_c, k))\&\mathbf{E}, \tag{5}$$

where & is the pixel-wise AND operator, and $dilation(\cdot, k)$ and $erosion(\cdot, k)$ are the morphological operations of image dilation and erosion in k pixels, respectively. Sample images can be found in Fig. 4.



(a) Image (b) Sobel edge (c) Coarse mask (d) Contours from coarse mask (e) Sparse contour map

Fig. 4. Supervision maps for contour refinement.

Second Stage Sparsely-Supervised Learning. To implement the supplement boundary supervision, we propose an additional contour-sensitive loss (i.e., $\mathcal{L}_{contour}$) to the existing losses to fine-tune the nucleus contours. Similarly, we also perform the supervision sparsely using our generated contour map. The contour-sensitive loss is defined as

$$\mathcal{L}_{contour}(\mathbf{W}) = \parallel ReLU(\mathbf{E}_r) \cdot (sobel(f(\mathbf{I})) - \mathbf{E}_r) \parallel_F^2, \tag{6}$$

Note that, the sobel operation is differentiable, and thus $\mathbf{W}$ can be optimized by backpropagation.

2.3 Implementation Details

During the whole training process, the segmentation model is a unified FCN of LinkNet [7], while different synergistic tasks with corresponding losses is applied to the same model output. Our model is implemented based on Keras with Tensorflow backend. The scaling parameter α is set to 0.05. The parameter k for morphological operations is set to 5.

In our weakly-supervised framework, we initialized the network with pre-trained parameters from an natural image segmentation dataset. Because of the lack of training samples, random cropping, scaling, rotation, flipping, brightness, and gamma transformation are utilized for data augmentation. We randomly crop the input image into the size of 512×512 for training the model. For every coarse segmentation iteration, we employ $\mathcal{L}_{point}$, $\mathcal{L}_{voronoi}$, and $\mathcal{L}_{polar}$ with weights of 1.0, 0.1, 0.1, respectively, to train network in 200 epochs. While in the contour estimation stage, we update the network by introducing an additional loss of $\mathcal{L}_{contour}$, to refine the model in 50 epochs. And the final loss weights are 0.01, 0.01, 0.01, 1.0, respectively. We employ Adam optimizer with a learning rate of 0.001 for both stages.

3 Experiments

3.1 Datasets

We evaluate our proposed weakly-supervised framework on two independent nucleus segmentation datasets: MoNuSeg [8] and TNBC [9]. MoNuSeg consists of 30 images of size 1000×1000, which are selected from the TCGA website of different cancer types from multiple hospitals. And TNBC is comprised of 50 images of size 512×512, which are extracted from slides of a cohort of Triple Negative Breast Cancer (TNBC) patients, scanned with Philips Ultra Fast Scanner 1.6RA. Both MoNuSeg and TNBC have pixel-level mask annotations. Therefore we can generate the points annotation for the training set by calculating the central point (with a random bias) of each nucleus mask. We adopt tenfold cross-validation for evaluation.

3.2 Evaluation Metrics

We use four metrics for evaluation, including two pixel-level criteria (i.e., pixel-level IoU and F1 score) and two object-level criteria (i.e., object-level Dice coefficient [10] and Aggregated Jaccard Index (AJI) [8]). The detailed definitions of these metrics are provided in [3,8]. Note that, the pixel-level F1 score is also known as the pixel-level Dice coefficient.

3.3 Results and Comparison

We compare our method with three weakly-supervised methods [1,3,5]. It should be noted that results from [3] are obtained by running the provided code, while

Table 1. Ten-fold validation results on MoNuSeg and TNBC datasets

MoNuSeg				
Methods	Pixel-level		Object-level	
	IoU	F1 score	Dice	AJI
Issam et al. [1]	0.5710 ± 0.02	-	-	-
Yoo et al. [5]	0.6136 ± 0.04	-	-	-
Qu et al. [3]	0.5789 ± 0.06	0.7320 ± 0.05	0.7021 ± 0.04	$\mathbf{0.4964 \pm 0.06}$
Our method	$\mathbf{0.6239 \pm 0.03}$	$\mathbf{0.7638 \pm 0.02}$	$\mathbf{0.7132 \pm 0.02}$	0.4927 ± 0.04
Fully supervised	0.6494 ± 0.04	0.7859 ± 0.02	0.7358 ± 0.03	0.5169 ± 0.05
TNBC				
Methods	Pixel-level		Object-level	
	IoU	F1 score	Dice	AJI
Issam et al. [1]	0.5504 ± 0.04	-	-	-
Yoo et al. [5]	0.6038 ± 0.03	-	-	-
Qu et al. [3]	0.5420 ± 0.04	0.7008 ± 0.04	0.6931 ± 0.04	0.5181 ± 0.05
Our method	$\mathbf{0.6393 \pm 0.03}$	$\mathbf{0.7510 \pm 0.04}$	$\mathbf{0.7413 \pm 0.03}$	$\mathbf{0.5509 \pm 0.04}$
Fully supervised	0.6950 ± 0.03	0.8022 ± 0.03	0.7881 ± 0.02	0.6233 ± 0.03

Table 2. Comparison of different iterations

MoNuSeg				
Iteration	Pixel-level		Object-level	
	IoU	F1 score	Dice	AJI
First-stage-r1	0.2315 ± 0.05	0.3710 ± 0.06	0.3771 ± 0.06	0.2164 ± 0.04
First-stage-r2	0.4198 ± 0.06	0.5864 ± 0.06	0.5741 ± 0.05	0.3727 ± 0.05
First-stage-r3	0.5244 ± 0.03	0.6860 ± 0.02	0.6497 ± 0.03	0.4348 ± 0.05
First-stage-r4	0.5080 ± 0.04	0.6704 ± 0.03	0.6175 ± 0.04	0.3907 ± 0.06
Second-stage-r1	$\mathbf{0.6239 \pm 0.03}$	$\mathbf{0.7638 \pm 0.02}$	$\mathbf{0.7132 \pm 0.02}$	$\mathbf{0.4927 \pm 0.04}$
TNBC				
Iteration	Pixel-level		Object-level	
	IoU	F1 score	Dice	AJI
First-stage-r1	0.3426 ± 0.05	0.5053 ± 0.06	0.4984 ± 0.06	0.3303 ± 0.05
First-stage-r2	0.4836 ± 0.08	0.6417 ± 0.08	0.6403 ± 0.05	0.4412 ± 0.07
First-stage-r3	0.5424 ± 0.06	0.6662 ± 0.07	0.6568 ± 0.05	0.4523 ± 0.05
First-stage-r4	0.4362 ± 0.09	0.5895 ± 0.10	0.5664 ± 0.10	0.3450 ± 0.09
Second-stage-r1	$\mathbf{0.6393 \pm 0.03}$	$\mathbf{0.7510 \pm 0.04}$	$\mathbf{0.7413 \pm 0.03}$	$\mathbf{0.5509 \pm 0.04}$

results for [1,5] are obtained from related paper [5]. Furthermore, we train a fully supervised model to illustrate the upper limit of our method. As shown in Table 1, in comparison with all weakly-supervised methods, our method almost achieves the best segmentation performance (except AJI on MoNuSeg set) on two datasets in terms of all evaluation criteria. Moreover, our method can achieve a competitive result compared with the fully supervised model.

To illustrate the effect of the point-to-region stage and contour-refine stage, Table 2 lists the results of each iteration. In the point-to-region stage, the accuracies of the first three iterations are gradually increased, while the fourth iteration decreases the performance. This is because when the positive segmentation results gradually reach the nucleus scale (even larger scale), certain negative pixels will be introduced into the positive point distance map according to Eq. 1, thus leading to unreliable positive map. However, the last row of Table 2 shows that after the contour-refinement stage, the segmentation model can better fit the nucleus edges, thereby further improving the effectiveness of the segmentation model.

4 Conclusion

In this paper, we propose a weakly-supervised segmentation framework based on point annotations. First, we train a sparse segmentation model through multiple iterations, and then we propose to use the additional contour-sensitive loss for contour refinement. In the experiments, our method can obtain a superior segmentation performance compared with the state-of-the-art weakly-supervised methods using point supervision. It suggests the effectiveness of our proposed coarse-to-fine learning framework.

References

1. Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the blobs: counting by localization with point supervision. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 560–576. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_34
2. Nishimura, K., Ker, D.F.E., Bise, R.: Weakly supervised cell instance segmentation by propagating from detection response. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11764, pp. 649–657. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32239-7_72
3. Qu, H., Wu, P., Huang, Q., et al.: Weakly supervised deep nuclei segmentation using points annotation in histopathology images. In: International Conference on Medical Imaging with Deep Learning, pp. 390–400 (2019)
4. Chamanzar, A., Nie, Y.: Weakly supervised multi-task learning for cell detection and segmentation. arXiv preprint arXiv:1910.12326 (2019)
5. Yoo, I., Yoo, D., Paeng, K.: PseudoEdgeNet: nuclei segmentation only with point annotations. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11764, pp. 731–739. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32239-7_81

6. Hershey, J.R., Chen, Z., Le Roux, J., et al.: Deep clustering: discriminative embeddings for segmentation and separation. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 31–35 (2016)
7. Chaurasia, A., Culurciello, E.: Linknet: exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP). IEEE, pp. 1–4 (2017)
8. Kumar, N., Verma, R., Sharma, S., et al.: A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE Trans. Med. Imaging **36**(7), 1550–1560 (2017)
9. Naylor, P., Lae, M., Reyal, F., et al.: Segmentation of nuclei in histopathology images by deep regression of the distance map. IEEE Trans. Med. Imaging **38**(2), 448–459 (2018)
10. Sirinukunwattana, K., Snead, D.R.J., Rajpoot, N.M.: A stochastic polygons model for glandular structures in colon histology images. IEEE Trans. Med. Imaging **34**(11), 2366–2378 (2015)
11. Sadanandan, S.K., Ranefall, P., Le Guyader, S., et al.: Automated training of deep convolutional neural networks for cell segmentation. Sci. Rep. **7**(1), 1–7 (2017)
12. Hatipoglu, N., Bilgin, G.: Cell segmentation in histopathological images with deep learning algorithms by utilizing spatial relationships. Med. Biol. Eng. Comput. **55**(10), 1829–1848 (2017)