Accurate Nuclear Segmentation with Center Vector Encoding

Jiahui Li, Zhiqiang Hu, and Shuang Yang
SenseTime Research
{lijiahui,huzhiqiang,yangshuang1}@sensetime.com

Abstract. Nuclear segmentation is important and frequently demanded for pathology image analysis, yet is also challenging due to nuclear crowd-edness and possible occlusion. In this paper, we present a novel bottom-up method for nuclear segmentation. The concepts of Center Mask and Center Vector are introduced to better depict the relationship between pixels and nuclear instances. The instance differentiation process are thus largely simplified and easier to understand. Experiments demonstrate the effectiveness of Center Vector Encoding, where our method outperforms state-of-the-arts by a clear margin.

1 Introduction

Pathology has long been regarded as the gold standard for medical diagnosis, especially cancer-related diagnosis. Nuclear segmentation is one of the most important and frequently demanded tasks of pathology image analysis, acting as the building block for nuclear statistical analysis such as size, density, counts, etc., which in turn contribute to cancer grading, therapy planning and outcome prediction. On the other hand, nuclear segmentation is laborious, tedious, and prone to errors for manual processing, and challenging for automatic methods due to nuclear crowdedness and possible occlusion.

Nuclear segmentation aims to segment out the nuclear areas from the backgrounds, as well as differentiate nuclear instances. The task can be well formulated as an instance segmentation problem. For the instance segmentation problem, there are two main frameworks, namely top-down and bottom-up. The top-down framework begins by localizing object instances, commonly in the form of bounding boxes, and then performs the segmentation within the bounding box. Mask R-CNN and its variants [9,13] are of the state-of-the-art top-down models. The bottom-up framework, on the contrary, performs the semantic segmentation on the whole image firstly, and differentiate instances with post-processing based on prior knowledge about the objects [3,5,19,8]. With the incorporation of prior knowledge, the bottom-up methods are challenging to design, but may perform better than general-purpose top-down counterparts.

In this work we focus on bottom-up methods. Among the previous works [1,2,6,10,11,12,16,17,18,20,21,22,23,24,25,26], Xing et al. [21] proposed a two-class Convolutional Neural Network (CNN) based method for the first-step semantic
segmentation that predicts inside and outside mask, and differentiate nuclear instances by carefully-designed distance transform and region growth. However, the method suffers from under-segmentation, especially for crowded nuclei. As a follow-up, Kumar et al. [11] extended the model to a three-class CNN, which predicts boundary mask in addition to inside and outside mask. They also proposed an anisotropic region growth method for instance differentiation, based on estimated probabilities in the three masks. Despite the decent improvement compared with Xing et al. [21], the Kumar et al. [11] method has at least three limitations:

1. A simple boundary mask takes limited advantage of overall information. For example, distance information is not explicitly modeled.
2. The boundary targets during training are sensitive to the nuclear annotations. Different annotators may deliver slightly different annotations even for the same nucleus, but in this case the boundaries will become totally different. Since the annotations can never be perfect, the sensitivity will make the model harder to learn.
3. The instance differentiation process is complex. Multiple thresholds are applied to control the region growth algorithm. The process behaves more like ad-hoc rules and is hard to tune and understand.

In this paper, we propose a novel method for nuclear segmentation that overcomes the aforementioned shortcomings. As shown in Fig. 1, our method begins by Fully Convolutional Neural Network (FCN) [14,15] for semantic segmentation. Instead of predicting boundaries, our method estimates the Center Mask, Center Vector, as well as the common Inside Mask. The Center Mask encodes the center regions of nuclei, the target of which during training is generated by morphological operations and distance thresholding. The Center Vector, on the other hand, encodes for each inside-nucleus pixel the relative displacement with respect to the centers. It contains two channels, for horizontal and vertical directions respectively. During inference, the center region of each nuclear instance is firstly derived by applying connected component analysis to the predicted Center Mask, and then each inside-nucleus pixel (predicted by Inside Mask after some processing) is assigned to a center region according to the predicted Center Vector. Pixels with Center Vector not pointing to any center region are assigned to the nearest center region. Finally we refine the Instance Mask, for example filling the holes, to remove the artifacts.

Compared with Kumar et al. [11], the concept of Center Vector in our method takes into consideration the distance with respect to the centers, instead of the dichotomous boundary labels, so we utilize more information for the model to separate touching nuclei. Also, our method is less sensitive to the annotations. For slighted perturbed annotations, only a tiny fraction of Center Vector targets will be affected, and the model can still learn well based on the majority of correct supervisions. Lastly, the Center Vector guides the pixels to find its center. The relationship between pixels and nuclear instances is clear with the concept of Center Vector. The process is straightforward, easier to understand and implement.
Fig. 1. (a) Overview of our framework: We apply Fully Convolutional Neural Network (FCN) for semantic segmentation and predicts Inside Mask, Center Mask and Center Vector, which are then utilized in instance differentiation to generate Instance Mask. (b) Illustration of Center Mask and Center Vector. Center Mask encodes the center region of nuclei, while Center Vector encodes the relative displacement of each inside-nucleus pixel with respect to the corresponding center.

We perform extensive experiments on the dataset released by Kumar et al. [11]. The quantitative results demonstrate the superiority of our method in performance. Our method outperforms state-of-the-arts by a clear margin. We also conduct ablation studies to investigate the benefits gained by introducing Center Vector, both in training phase and in inference phase. Finally, we show qualitative results to give an insight why the proposed pipeline is better than a boundary-prediction method.

Our contributions are summarized as follows:

1. We introduce the concepts of Center Mask and Center Vector to better depict the relationship between pixels and nuclear instances.
2. Based on the Center Vector Encoding, we present a pipeline for nuclear segmentation, easy to understand and implement.
3. Our model achieves state-of-the-art performance in the challenging dataset released by Kumar et al. [11]. Besides, experiments demonstrate the benefits of Center Vector for better instance differentiation.

2 Center Vector Encoding

As shown in Fig. 1(a), the entire pipeline consists of two parts: semantic segmentation and instance differentiation. We apply Fully Convolutional Neural Network (FCN) [14] for semantic segmentation and predicts Inside Mask, Center Mask and Center Vector, which are then utilized in instance differentiation to generate Instance Mask.

2.1 Center Mask and Center Vector in Semantic Segmentation

Center Mask and Center Vector are the core concepts of our work. The Center Mask encodes the center regions of nuclei. The target center region of a nuclear instance during training is calculated as follows: The instance mask for the nuclear instance is firstly processed with morphological erosion, and then we calculate the distance from each pixel to the boundary in the eroded mask, and take as the center region those pixels with distance larger than a threshold. The operations mainly aim to ensure center regions of touching nuclei to separate.

The Center Vector, on the other hand, encodes the relative displacement of each inside-nucleus pixel with respect to the corresponding center. The center $C_i$ for a nuclear instance $i$ is defined as the geometric center of the instance mask, and is denoted as $C_i = (\tilde{X}_i, \tilde{Y}_i)$. For each pixel $P_{ij} = (X_{ij}, Y_{ij})$ within the instance $i$, the Center Vector $\Delta P_{ij}$ of the pixel is defined as

$$\Delta P_{ij} = P_{ij} - C_i = (X_{ij} - \tilde{X}_i, Y_{ij} - \tilde{Y}_i)$$

(1)

For pixels not belonging to any nuclear instance, i.e. the backgrounds, the Center Vector is not defined, and the corresponding loss is ignored during training. See Fig. 1(b) for an illustration of the concepts of Center Mask and Center Vector.

Compared with boundary encoding [11], the Center Vector encodes the distance between pixels and centers instead of the dichotomous boundary labels, and thus contains richer information. The Center Vector is smooth inside a nuclear instance, making it easier for the model to learn, but changes sharply in the boundary areas, especially in the boundary of touching nuclei, forcing the model to pay more attention to the boundaries. Moreover, Center Vector is less sensitive to the annotation perturbation. Even if annotation areas are slightly enlarged or reduced, for example due to different annotation styles by different annotators, only a tiny fraction of Center Vector targets will be affected, and the model can still learn well based on the majority of correct supervision.

Finally, the Inside Mask simply contains all the foreground pixels – pixels within the overall region of some nuclear instance. For an input image, the Inside Mask and Center Mask are of the same shape as the image, with 1 channel each
in the semantic segmentation output; Center Vector contains 2 channels for horizontal and vertical directions respectively, and is also of the same shape as the input image.

2.2 Instance Differentiation

During inference, the semantic segmentation model outputs three kind of prediction: Inside Mask, Center Mask and Center Vector, from which we will derive the final Instance Mask. To this end, we firstly perform the connected component analysis to the Inside Mask, and remove regions which have no intersection with Center Mask. This serves as a kind of false positive suppression. On the other hand, we also perform the connected component analysis to the Center Mask, and takes each resulting region as the center region of one nuclear instance. Finally we will assign pixels in the false-positive-suppressed Inside Mask to the center regions: Pixels in the Center Mask are directly assigned to the corresponding center region without considering the Center Vector; For pixels not in the Center Mask but in the Inside Mask, we assign them to the center region their Center Vector points to, or the nearest center region in case that their Center Vector does not point to any valid center region. Minor refinement are applied afterwards, for example filling the holes, to remove the artifacts.

With the concepts of Center Mask and Center Vector, the instance differentiation process is straightforward and easy to understand. To summarize, the Center Mask represents the center region of each nuclear instance, while the Center Vector links the pixels to the corresponding center regions.

3 Experiments

3.1 Implementations

It is worthy noting that our method does not rely on any particular network architecture for the semantic segmentation model. For implementation we use Deep Layer Aggregation (DLA) [23] for the network architecture in this paper. DLA extends common network structures with deep aggregations. The term deep aggregation refers to aggregations that are nonlinear, compositional and going through multiple stages. DLA introduces two types of deep aggregations: Iterative Deep Aggregation and Hierarchical Deep Aggregation. Iterative Deep Aggregation merges layers iteratively, while Hierarchical Deep Aggregation aggregates layers in a tree-like hierarchical manner. With these two types of deep aggregations, the network can better fuse information from multiple layers and scales, and thus achieves better performance in various classification and segmentation problems. Fig. 2 illustrates the Iterative Deep Aggregation and Hierarchical Deep Aggregation introduced by DLA.

For the loss function, we utilize a combination of pixel-wise cross entropy (CE) loss and Intersection-Over-Union (IOU) loss for the optimization of Inside Mask (IM) and Center Mask (CM) estimation. For Center Vector (CV, with
Fig. 2. Illustration of the Iterative Deep Aggregation and Hierarchical Deep Aggregation introduced by Deep Layer Aggregation. Iterative Deep Aggregation merges layers iteratively, while Hierarchical Deep Aggregation aggregates layers in a tree-like hierarchical manner.

CV_X and CV_Y for two directions) with continuous target values, we apply pixel-wise mean square (MS) loss. Please see (2), (3), and (4) for detailed formulations, where \( y_i \) and \( p_i \) are targets and predictions for pixel \( i \) respectively, i.e., class labels/probabilities for CE and IOU loss but distance targets/estimations for MS loss. Finally, the total loss is a weighted summation of all the aforementioned losses, as shown in (5), where \( \alpha, \beta, \gamma \) are the balancing parameters.

\[
l_{CE} = \sum_{S \in \{IM, CM\}} \sum_{i \in S} y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \tag{2}
\]

\[
l_{IOU} = \sum_{S \in \{IM, CM\}} 1 - \frac{\sum_{i \in S} y_i p_i}{\sum_{i \in S} y_i + \sum_{i \in S} p_i - \sum_{i \in S} y_i p_i} \tag{3}
\]

\[
l_{MS} = \sum_{S \in \{CV_X, CV_Y\}} \sum_{i \in S} (p_i - y_i)^2 \tag{4}
\]

\[
l = \alpha l_{CE} + \beta l_{IOU} + \gamma l_{MS} \tag{5}
\]

We use DLA-34 for the model. The model is implemented in Pytorch 0.4. We train the model for 2000 epochs. Using SGD as the optimizer, learning rate is initially set as 0.01 and decayed polynomially. Batchsize is 4 and weight decay is 0.0001. In loss function we set \( \alpha = 10, \beta = 10, \gamma = 1 \) to balance different losses.

### 3.2 Dataset and Evaluation Metric

We perform extensive experiments on the dataset released by Kumar et al. [11] to evaluate the proposed method. The dataset consists of 30 Hematoxylin and Eosin (H&E) stained images, all of which are of the size 1000 × 1000 pixels. Nuclear instances are carefully annotated so as to generate the instance mask for each image. The images come from multiple organs: 4 organs (breast, kidney, liver and prostate) have 6 images each, while another 3 organs (bladder, colon and stomach) have 2 images each. Following the same principle as Kumar et al. [11], that is test data must contain images from organs never seen in train data,
we sample 4 images from each of 4 majority-class organs (breast, kidney, liver and prostate), a set of 16 images in total for training and put the rest for test. The multi-organ diversity and zero-shot setting make the problem even more challenging, but closer to clinical practice.

The performance of nuclear segmentation solutions is evaluated based on the metric Aggregated Jaccard Index (AJI) [11]. AJI penalizes both segmentation errors and instance differentiation errors, and is a balancing metric for a comprehensive evaluation. We summarize the AJI calculation steps listed in [11] as the equation (6), where \( \{G_i\}_{i=1}^M \) and \( \{P_j\}_{j=1}^N \) denote the set of annotated nuclei and predicted nuclei, respectively; \( f(i) = \arg\max_j |G_i \cap P_j|/|G_i \cup P_j| \) assigns best-match prediction to each annotated nuclear instance; \( [N] = \{1,2,\ldots,N\} \) and \( f([N]) = \{f(1),f(2),\ldots,f(N)\} \) are convenient notations.

\[
\text{AJI} = \frac{\sum_{i=1}^M |G_i \cap P_{f(i)}|}{\sum_{i=1}^M |G_i \cup P_{f(i)}| + \sum_{j \in [N] \setminus f([M])} |P_j|}
\]

\[
\text{IOU} = \frac{|G \cap P|}{|G \cup P|}
\]

\[
\text{Dice} = \frac{2|G \cap P|}{|G| + |P|}
\]

To investigate the error modes, we also apply two metrics originally designed for the semantic segmentation task: Intersection-Over-Union (IOU) and Dice. Please refer to (7) and (8) for detailed formulations, where \( G = \bigcup_{i=1}^M G_i \) and \( P = \bigcup_{j=1}^N P_j \) are the global annotations and predictions, respectively. Note that IOU is the upper bound of AJI: We have

\[
\sum_{i=1}^M |G_i \cap P_{f(i)}| \leq |G \cap P|
\]

and

\[
\sum_{i=1}^M |G_i \cup P_{f(i)}| + \sum_{j \in [N] \setminus f([M])} |P_j| \geq |G \cup P|
\]

leading to

\[
\text{AJI} \leq \frac{\sum_{i=1}^M |G_i \cap P_{f(i)}|}{\sum_{i=1}^M |G_i \cup P_{f(i)}| + \sum_{j \in [N] \setminus f([M])} |P_j|} \leq \frac{|G \cap P|}{|G \cup P|} = \text{IOU}
\]

Dice is yet the upper bound of IOU. As a result, the metrics IOU and Dice can be used for distinguishing instance differentiation errors from segmentation errors.

3.3 Results

In Table 1 we show the performance comparison of our method against state-of-the-art methods: CNN2 [21] for two-class CNN (Inside, Outside) and CNN3 [11]
Table 1. Performance comparison of our method against state-of-the-art methods: CNN2 [21] for two-class CNN (Inside, Outside) and CNN3 [11] for three-class CNN (Inside, Outside, Boundary).

| Organs  | AXI | Dice |
|---------|-----|------|
|         | CNN2 | CNN3 | ours | CNN2 | CNN3 | ours |
| Breast  | 0.425 | 0.538 | 0.601 | 0.646 | 0.718 | 0.819 |
| Liver   | 0.371 | 0.516 | 0.601 | 0.610 | 0.688 | 0.797 |
| Kidney  | 0.407 | 0.508 | 0.487 | 0.656 | 0.722 | 0.737 |
| Prostate| 0.228 | 0.573 | 0.615 | 0.704 | 0.792 | 0.792 |
| Bladder | 0.319 | 0.437 | 0.557 | 0.715 | 0.781 | 0.815 |
| Colon   | 0.308 | 0.521 | 0.452 | 0.664 | 0.739 | 0.745 |
| Stomach | 0.379 | 0.446 | 0.612 | 0.855 | 0.895 | 0.848 |
| Overall | 0.348 | 0.508 | 0.561 | 0.693 | 0.762 | 0.793 |

Table 2. Results of ablation studies. Center Vector is not utilized in V1, only during training in V2, and both during training and inference in V3. We add results from CNN3 [11] for comparison.

| Organs  | AXI | IOU | Dice |
|---------|-----|-----|------|
|         | CNN3 | V1  | V2  | V3  | CNN3 | V1  | V2  | V3  |
| Breast  | 0.538 | 0.597 | 0.595 | 0.601 | None | 0.699 | 0.677 | 0.694 | 0.718 | 0.822 | 0.807 | 0.819 |
| Liver   | 0.516 | 0.598 | 0.573 | 0.601 | None | 0.672 | 0.646 | 0.663 | 0.688 | 0.803 | 0.784 | 0.797 |
| Kidney  | 0.508 | 0.496 | 0.483 | 0.487 | None | 0.606 | 0.574 | 0.585 | 0.722 | 0.754 | 0.727 | 0.737 |
| Prostate| 0.573 | 0.600 | 0.626 | 0.615 | None | 0.645 | 0.648 | 0.655 | 0.792 | 0.784 | 0.786 | 0.792 |
| Bladder | 0.437 | 0.504 | 0.544 | 0.557 | None | 0.673 | 0.677 | 0.693 | 0.781 | 0.799 | 0.803 | 0.815 |
| Colon   | 0.521 | 0.441 | 0.437 | 0.452 | None | 0.614 | 0.560 | 0.594 | 0.739 | 0.763 | 0.717 | 0.745 |
| Stomach | 0.446 | 0.558 | 0.593 | 0.612 | None | 0.734 | 0.717 | 0.736 | 0.895 | 0.846 | 0.835 | 0.848 |
| Overall | 0.508 | 0.542 | 0.550 | 0.561 | None | 0.664 | 0.663 | 0.663 | 0.762 | 0.794 | 0.791 | 0.793 |

for three-class CNN (Inside, Outside, Boundary). We omit IOU scores since they are not reported in the literature. From the comparison it is clear to see that our method outperforms the state-of-the-arts by a large margin, both with respect...
to the instance-level AJI metric and with respect to the segmentation-level Dice metric.

We also perform ablation studies to investigate the benefits gained from the design. We train the semantic segmentation model to predict Inside Mask and Center Mask only, and differentiate instances by Random Walker [7] seeded with center regions derived from Center Mask predictions. We term this setting as V1. Further for the setting V2, we add Center Vector supervisions during training, but we ignore the Center Vector output during inference and also apply Random Walker [7]. In this case, Center Vector serves as a way to facilitate training. Finally the setting V3 is the entire method we describe above, where Center Vector is utilized both in training and during inference.

Table 2 shows the results of ablation studies. We add results from CNN3 [11] for comparison. The performance gain of V1 from CNN3 [11] mainly attributes to the better Deep Layer Aggregation (DLA) [23] architecture. Comparing V1, V2 and V3, the AJI score is increasing, which demonstrates the effectiveness of Center Vector, both in training and during inference. However, the IOU and Dice scores are largely comparable for V1, V2 and V3, showing that Center Vector works mainly from suppressing errors in the instance differentiation step.

We also display some qualitative results to give an insight why Center Vector is effective, as shown in Fig. 3 and Fig. 4. Fig. 3 compares qualitative results from V1 and V2. The V2 model, with Center Vector during training, learns to better separate touching nuclei in the Center Mask, showing that Center Vector supervision guides the model to concentrate more in the center regions and learn better Center Mask estimation. On the other hand, Fig. 4 compares V2 with V3, where the models differ only in the instance differentiation step. It can be seen that Random Walker method in V2 tends to separate touching nuclei

![Fig. 3. Qualitative results from V1 and V2: The V2 model, with Center Vector during training, learns to better separate touching nuclei in the Center Mask.](image)
Fig. 4. Qualitative results from V2 and V3: Random Walker method in V2 tends to separate touching nuclei with a “straight cut” while Center Vector generates more natural and realistic boundaries.

with a “straight cut” while Center Vector generates more natural and realistic boundaries.

4 Conclusion

We present a novel bottom-up method for nuclear segmentation. The concepts of Center Mask and Center Vector are introduced to better depict the relationship between pixels and nuclear instances. Based on the Center Vector Encoding, we develop a pipeline for nuclear segmentation, easy to understand and implement. Experiments demonstrate the effectiveness of Center Vector Encoding, where our method outperforms state-of-the-arts by a clear margin.

References

1. Arbelle, A., Raviv, T.R.: Microscopy cell segmentation via adversarial neural networks. In: Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on. pp. 645–648. IEEE (2018)
2. Bamford, P.: Automating cell segmentation evaluation with annotated examples. In: APRS Workshop on Digital Image Computing. pp. 21–25 (2003)
3. Belsare, A., Mushrif, M., Pangarkar, M.: Breast epithelial duct region segmentation using intuitionistic fuzzy based multi-texture image map. In: 2017 14th IEEE India Council International Conference (INDICON). pp. 1–6. IEEE (2017)
4. Chen, L.C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., Adam, H.: Masklab: Instance segmentation by refining object detection with semantic and direction features. arXiv preprint arXiv:1712.04837 2 (2018)
5. Cui, Y., Zhang, G., Liu, Z., Xiong, Z., Hu, J.: A deep learning algorithm for one-step contour aware nuclei segmentation of histopathological images. CoRR abs/1803.02786 (2018). http://arxiv.org/abs/1803.02786
6. Fu, D., Xie, X.S.: Reliable cell segmentation based on spectral phasor analysis of hyperspectral stimulated raman scattering imaging data. Analytical chemistry 86(9), 4115–4119 (2014)
7. Grady, L.: Random walks for image segmentation. IEEE transactions on pattern analysis and machine intelligence 28(11), 1768–1783 (2006)
8. Gurcan, M.N., Boucheron, L., Can, A., Madabhushi, A., Rajpoot, N., Yener, B.: Histopathological image analysis: A review. IEEE reviews in biomedical engineering 2, 147 (2009)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. pp. 2980–2988. IEEE (2017)
10. Ho, D.J., Fu, C., Salama, P., Dunn, K.W., Delp, E.J.: Nuclei segmentation of fluorescence microscopy images using three dimensional convolutional neural networks (2017)
11. Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., Sethi, A.: A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE transactions on medical imaging 36(7), 1550–1560 (2017)
12. Li, G., Liu, T., Tarokh, A., Nie, J., Guo, L., Mara, A., Holley, S., Wong, S.T.: 3d cell nuclei segmentation based on gradient flow tracking. BMC Cell Biology 8(1), 40 (Sep 2007). https://doi.org/10.1186/1471-2121-8-40 https://doi.org/10.1186/1471-2121-8-40
13. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. CoRR abs/1803.01534 (2018), http://arxiv.org/abs/1803.01534.
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
15. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597 (2015), http://arxiv.org/abs/1505.04597
16. Sadanandan, S.K., Ranefall, P., Le Guyader, S., Wühlby, C.: Automated training of deep convolutional neural networks for cell segmentation. Scientific reports 7(1), 7860 (2017)
17. Stegmaier, J., Spina, T.V., Falcão, A.X., Bartschat, A., Mikut, R., Meyerowitz, E., Cunha, A.: Cell segmentation in 3d confocal images using supervoxel merge-forests with cnn-based hypothesis selection. In: Biomedical Imaging (ISBI 2018), 2018 IEEE 15th International Symposium on. pp. 382–386. IEEE (2018)
18. Su, H., Yin, Z., Huh, S., Kanade, T.: Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features. Medical image analysis 17(7), 746–765 (2013)
19. Wang, P., Hu, X., Li, Y., Liu, Q., Zhu, X.: Automatic cell nuclei segmentation and classification of breast cancer histopathology images. Signal Processing 122, 1–13 (2016)
20. Wang, Z., Li, H.: Generalizing cell segmentation and quantification. BMC bioinformatics 18(1), 189 (2017)
21. Xing, F., Xie, Y., Yang, L.: An automatic learning-based framework for robust nucleus segmentation. IEEE transactions on medical imaging 35(2), 550–566 (2016)
22. Yin, Z., Bise, R., Chen, M., Kanade, T.: Cell segmentation in microscopy imagery using a bag of local bayesian classifiers. In: Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on. pp. 125–128. IEEE (2010)
23. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. arXiv preprint arXiv:1707.06484 (2017)
24. Zhang, X., Liu, W., Dundar, M., Badve, S., Zhang, S.: Towards large-scale histopathological image analysis: Hashing-based image retrieval. IEEE Transactions on Medical Imaging 34(2), 496–506 (2015)
25. Zhang, X., Xing, F., Su, H., Yang, L., Zhang, S.: High-throughput histopathological image analysis via robust cell segmentation and hashing. Medical image analysis \textbf{26}(1), 306–315 (2015)

26. Zhou, Y., Kuijper, A., Heise, B., He, L.: Cell segmentation using level set method. na (2007)