CSU-CN

Authors: Xuhua Yan

Email: 0921160212@csu.edu.cn

Platform: Linux

Prerequisites: Python 3.6 with Tensorflow

CSU-CN: SUMMARY

I only used one fully convolution network (FCN) [1] which outputs two maps. One output is a normalized distance map and the other is a seed map. Then, I applied seed-based watershed transform to achieve cell instance segmentation.

CSU-CN: PREPROCESSING

Let S be a dataset, where S={(x_i, g_i)} is the subset of images x_i with instance annotation g_i taken from the available gold truth and silver truth. To solve the instance segmentation task, I used a fully convolution network to transform g_i into two semantic annotations D_i, M_i. D_i is the normalized Euclidean distance transform map for each cell. The distance values within each cell are normalized to range between 0 and 1. As for M_i, I calculate the centroid for each cell and generate a small disk around the centroid. The radius r_i of each disk is dependent on two parameters: p and r_0.

$$r_i = \max\left(\sqrt{p \cdot \frac{S_i}{\pi}}, r_0\right)$$

where S_i is the cell area, and r_0 is the lower bound of radius for every disk. In M_i, each pixel within a disk is set to 1, otherwise 0. I adjust the parameters p, r_0 to ensure that there is not overlap in M_i. For some datasets with low image contrast, I apply Contrast Limited Adaptive Histogram Equalization (CLAHE) and normalize the input images to range between 0 and 1.

CSU-CN: SEGMENTATION

I observed that many watershed-based deep learning methods obtain the seed map (M_i) in an indirect manner and there was a big difference between the number of generated seeds and the real number. So I use a fully convolution network which takes the U-Net [2] as the backbone to directly predict the seed map from the input image, which can reduce the noise introduced by extra processing. And to obtain the distance map (D_i) which is necessary for watershed transform, I add another output head to predict the normalized Euclidean distance map. These two output heads are added directly after the decoder of U-Net. As for training, I used shifting, scaling, rotating, and adding gaussian noise for the data augmentation. I use binary cross-entropy for the distance output head and weighted mean squared error (MSE) for the seed output head. The reason why I choose this loss combination is it can be easily extended to simultaneous cell instance segmentation and classification. As for the weights in MSE loss, I fix them to 0.5 and 10 for all datasets and 0.5 corresponds to weight of background pixels. I train a model on each dataset for 200 epochs with a batch size of 4. Adam optimizer with 3e-5 or 5e-5 initial learning rate are used.

CSU-CN: POST-PROCESSING

I apply thresholding with value of T_1 to the distance prediction to get the binarized segmentation map, value of T_2 to the seed prediction to get the seed map. Finally, I use seed-based watershed transform which takes the binarized mask, distance prediction and seed map as inputs to finish instance segmentation.

REFERENCES

1. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431-3440 (2015).
2. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention*, 234-241 (2015).