AU-TR

Authors: Mehmet Emin Bakır, Hacer Yalim Keles

Email: mebakir@ankara.edu.tr, hacer.keles@ankara.edu.tr

Platform: Linux

Prerequisites: Python 3 with Keras

AU-TR: SUMMARY

We used two cascaded convolutional neural networks. First network focuses on detecting cell instances. Second network predicts segmented cells and borders by using first network output as prior information. Then, we applied watershed segmentation to obtain the final result.

AU-TR: PREPROCESSING

We used silver truth annotations. However, we completed missing cells to avoid uncertainty. Also, we divided each label into three classes: background, cell and border. Background class is the same with silver truth. Cells are eroded by *E* pixels from the silver truth cells. Eroded regions are marked as borders. In the first network, we used only background and cell classes. In the second network, all three classes are used. We also applied histogram equalization to all input images and normalized them to range between 0 and 1.

AU-TR: SEGMENTATION

We observed that prior information about cell locations increases segmentation accuracy, so we used two different networks. In both networks, we used U-net [1] architecture. Firstly, we use a convolutional network to predict different cell regions from the input image. Then, we applied thresholding with the value of T_1 to the first network result. Then, by using both the input image and its first network output, we trained the second network to predict cells and borders. Thereby, we got more accurate results. As for data augmentation, we applied mirroring, rotating, shearing and zooming. We used a focal loss function [2] to handle data imbalance problem in training. We calculated focal loss weighting factors according to the ratio of the average number of pixels in each class. Our networks were trained for 300 epochs with a batch-size of 8. Adam optimizer with 0.0001 initial learning was used. Learning rate was reduced by *R* when 10 consecutive epochs had no improvement.

AU-TR: POST-PROCESSING

The second network produces images with three class outputs. We applied thresholding with the value of T_2 to these outputs. To obtain final segmentation, we used watershed method; the combination of cell and border labels are used as watershed input, cell is used as marker.

REFERENCES

1. 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).
2. Lin T, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**, 318-327 (2018).