BFR-GE (1)

Authors: Nils Körber

Email: nils.koerber@bfr.bund.de

Platform: Windows, Linux

Prerequisites: Python 3

BFR-GE (1): SUMMARY

Our approach utilizes the Microscopic Image Analyzer (MIA) software for image segmentation [1]. MIA is custom written software application allowing image labeling, neural network training, model predictions and post-processing.

BFR-GE (1): PREPROCESSING

All training input images were normalized to the range of [-1, 1].

BFR-GE (1): SEGMENTATION

For each dataset, a DeepLabv3+ [3] with Xception [4] backbone model was trained with MIA. All models were trained with 512 × 512 image patches that are randomly sampled from the training images. The Adam optimizer [2] with an initial learning rate of 0.0005, halved every 50 epochs, was used to optimize the cross entropy cost function. The same augmentation strategy was used for all trainings using image flipping, rotation, up to 10 % shearing, 90-110 % scaling and a 15 % probability of image blurring, piecewise affine image transformation or image dropout. The CTC labels were converted to MIA compatible labels. Since MIA counts touching objects as one, touching objects were separated by introducing a zero-valued boundary between them. Pixels between objects were weighted with a maximum weighting of w_{max} and a border distance parameter of d_{border} (as calculated in [5]).

For **Fluo-N2DH-SIM+**, our model was trained from scratch with a batch size of 16 for 500 epochs using the provided ground-truth reference segmentation annotations. The target cell class was weighted with 0.6 and background with 0.4. For **Fluo-N2DL-HeLa**, our model with pre-trained weights was trained with a batch size of 16 for 300 epochs using manually curated silver-truth reference segmentation annotations and the class weighting of 0.9 to 0.1.

All model predictions were done with test time augmentation, meaning that the input image was flipped and rotated by +-90 degree to generate a total of six input images. The model predictions of these transformed images were averaged to the final prediction.

BFR-GE (1): POST-PROCESSING

Post-processing was either omitted or the detected objects were split based on watershed transform using the predicted probabilities with a minimum object distance d_{min}. Upon conversion to the CTC labels, the detected objects were extended to neighboring objects to revert the zero-valued boundary.

REFERENCES

1. Nils Körber. MIA: An open source standalone deep learning application for microscopic image analysis. bioRxiv:2022.01.14.476308, 2022.
2. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2014.
3. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 801-818 (2018).
4. Chollet F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251-1258 (2017).
5. 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).