**FR-Fa-GE**

Authors: Thorsten Falk, Dominic Mai, Robert Bensch, Alexander Dovzhenko, Klaus Palme, Thomas Brox, Olaf Ronneberger

Email: falk@informatik.uni-freiburg.de

Platform: Linux

Prerequisites: MATLAB 2014b (x64)

*FR-Fa-GE: SUMMARY*

Our approach is a variant of **FR-Ro-GE** using the U-Net neural network architecture. Instead of individual U-Nets per challenge dataset, we trained one U-Net for general cell segmentation using annotated cell images from various modalities and species. We included the challenge training sets **Fluo-C2DL-MSC**, **Fluo-N2DH-GOWT1**, **Fluo-N3DH-SIM+** (subset), **Fluo-N2DL-HeLa**, **DIC-C2DH-HeLa**, **PhC-C2DH-U373**, and **PhC-C2DL-PSC** and four custom datasets into training. Three of these contain cells recorded with brightfield illumination (Pollen, protoplasts and microspores) to also capture this modality. The fourth is an additional PhC dataset showing HKPV cells. We use the same greedy label propagation as in **Fr-Ro-GE** for tracking.

*FR-Fa-GE: PREPROCESSING*

For each image, intensities were normalized to the [0, 1] range and images rescaled to a pixel size of 0.5×0.5µm.

*FR-Fa-GE: SEGMENTATION*

The segmentation is performed by the U-Net. It consists of a contracting path with a series of convolution, ReLU and max-pooling layers and an expansion path with a series of up-convolution, ReLU and convolution layers. In the expansion path, feature maps from the contracting path with the same resolution are copied (see [1] for the detailed architecture). For the challenge contribution, we averaged the predicted segmentation maps of the input image and its mirrored versions.

*Training the Segmentation Network*. The loss function for training is computed by a pixel-wise soft-max over the final feature maps combined with a weighted cross entropy loss function [1]. We pre-compute the loss-weight map for each ground truth segmentation map to compensate the unbalanced classfrequency in the training data set, and to force the network to learn the small separation borders, that we introduce between touching cells. The loss-weight map is then computed as

$$w(x) = w_c + w_0 \cdot \exp\left(-(d_1(x) + d_2(x))^2/(2\sigma_0^2)\right) + (1 - w_c) \cdot \exp\left(-(d_1(x) + d_2(x))^2/(2\sigma_1^2)\right) ,$$

where $w_c$ is a constant weight to approximately balance the class frequencies, $d_1{:}\Omega\rightarrow R$ denotes the distance to the border of the nearest cell and $d_2{:}\Omega\rightarrow R$ the distance to the border of the second nearest cell. In our experiments, we set $w_c = 0.1$, $w_0 = 50$, $\sigma_0 = 3$µm, and $\sigma_1 = 5$µm. The provided manual segmentation masks in the training data set do not cover all visible cells. To obtain a consistent background training set, we manually created “ignore” - regions that cover all unlabeled cells, and set the loss-weights to zero within these regions. Data augmentation is essential to teach the network the desired invariance and robustness properties, when only few training samples are available. For microscopic images, we primarily need shift and rotation invariance and robustness to deformations and gray value variations. Especially, random elastic deformations of the training samples seem to be the key concept to train a segmentation network with a very low number of annotated images. We generate smooth deformations using random displacement vectors on a coarse grid with grid point spacing of 150 pixels in both directions. The displacements are sampled from a Gaussian distribution with a standard deviation of 10 pixels per vector component. Dense per-pixel displacements are then computed using bicubic interpolation. Gray values of the input images are transformed using a random strictly increasing curve of which start point, end point and central curve slope are drawn from uniform distributions with ranges [-0.05, 0.05], [0.95, 1.05], and [0.8, 1.2] respectively. Intermediate values are determined using spline interpolation. Drop-out layers at the end of the contracting path perform further implicit data augmentation. The augmented input images and their corresponding segmentation maps are used to train the network with the ADAM implementation of Caffe. Due to the unpadded convolutions, the input image is larger than the output by a constant border width. To minimize the overhead and make maximum use of the GPU memory, we favor large input tiles over a large batch size and hence reduce the batch to a single image. To compensate for instable gradients, we accordingly set high momentum parameters (momentum=0.9, momentum2=0.999) such that a large number of the previously seen training samples determine the current optimization step. We trained the network with a fixed learning rate of 0.00001 for 150000 iterations. We overcome the imbalance in the numbers of images and annotated objects per dataset, by randomly drawing the presented patch at each iteration: we first randomly draw the dataset, then the image (both from uniform distributions) and finally the training tile to crop from the image from a distribution that favors foreground pixels as patch centers (factor of 10).

The output segmentation masks have a pixel size of 0.5×0.5µm. We upscale them to original image resolution using nearest neighbor interpolation.

### *FR-Fa-GE: TRACKING*

For tracking, we use a greedy algorithm. Each segment in frame t propagates its label to that segment in frame t + 1 with the highest overlap (measured as intersection over union). If a segment in frame t + 1 receives multiple labels, it prefers the segment in frame t with the highest overlap and discards the other labels. If a segment receives no label, a new label is assigned.

### *FR-Fa-GE: POST-PROCESSING*

No post-processing is carried out after tracking.

## 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).