KIT-Sch-GE (1)

Authors: Katharina Löffler, Tim Scherr, Ralf Mikut

Email: ralf.mikut@kit.edu

Platform: Linux (tested on Ubuntu 16.04 and 18.04)

Prerequisites: 32 GB RAM, 12 GB VRAM (24 GB VRAM for **Fluo-N3DL-TRIF**)

KIT-Sch-GE (1): SUMMARY

Our cell tracking method is a tracking by detection method split into a segmentation step and a tracking step which includes the correction of some segmentation errors. For the segmentation, a deep learning-based prediction of cell distance maps is used followed by a watershed post-processing. The tracking is based on a displacement estimation in combination with a matching formulated as maximum flow minimum cost problem.

KIT-Sch-GE (1): PREPROCESSING

This step involves a minimum-maximum normalization into the [-1, 1] range (whole volume for 3D data). For **Fluo-N3DL-TRIC**, the contrast limited adaptive histogram equalization (CLAHE) is applied to each slice independently. The model inputs are zero-padded if necessary. The frames of **Fluo-C3DH-H157** are downsampled to half of the original size.

KIT-Sch-GE (1): SEGMENTATION

The segmentation is based on a 2D convolutional neural network which is applied slice-wise to the 3D datasets. Our goal is to train a single model which can be used for all cell types. Only some parameters in the post-processing are adjusted to single datasets.

Architecture. Our architecture is based on the 2D U-Net architecture [1]. However, instead of using a single decoder path, two parallel decoder paths are used which allow each decoder path to focus on features related to the desired output. The maximum pooling layers are replaced with 2D convolutional layers with stride 2 and kernel size 3. Additionally, batch normalization layers are added. The number of feature maps is doubled from 64 feature maps to a maximum of 1024 and halved in each decoder path correspondingly. The rectified linear unit activation function is used within the network and a linear activation for the output layer.

Training Data. The assembled training dataset consists of the provided Cell Tracking Challenge training data, publicly available data, and synthetic data. Since some of the Cell Tracking Challenge gold truth data show only some annotated cells, and the silver truth data are not error free in any case, only selected 2D training data and slices of the 3D data are used as follows:

- **BF-C2DL-HSC**: 33 gold truth frames
- **BF-C2DL-MuSC**: 25 gold truth frames
- **Fluo-N2DH-GOWT1**: 4 gold truth frames, 20 silver truth frames
- **Fluo-N2DH-SIM+**: 35 gold truth frames
- **Fluo-N2DL-HeLa**: 40 silver truth frames
- **Fluo-N3DH-SIM+**: 10 gold truth slices
- **Fluo-N3DH-CE**: 6 gold truth slices
- **Fluo-C3DL-MDA231**: 6 gold truth slices
- Synthetic data: 30 slices
- BBBC038: 6 Drosophila slices
- SBDE4 in [2]: 24 slices

The final training dataset consists of crops of these images, with the size of 256×256 pixels. Crops showing less than four objects are removed from the training dataset automatically (not for **BF-C2DL-MuSC** and **Fluo-N3DH-CE**). In addition, some crops were removed manually, e.g. crops showing only parts of cells and many **Fluo-N2DL-HeLa** crops showing nearly the same. In total, the training dataset consist of 797 crops and the validation dataset of 200 crops.

Training Data Representation. A key for the successful application of supervised deep learning methods in the absence of large training data sets, is to learn outputs which generalize well. Therefore, instead of the often used cell and cell border representations (to get markers for a marker-based watershed post-processing for instance segmentation), distance transforms are used. In principle, this distance transform is sufficient to get markers for the post-processing. However, to reduce the probability to merge cells also cell neighbor distance transforms are used (with gray-scale closing) for the second decoder path of our architecture.

Training Process Settings. Models are trained for a maximum of 250 epochs with a batch size of 8 using the Adam optimizer with a starting learning rate of 0.0008. After 12 epochs without validation loss improvement, the learning rate is multiplied with 0.25 until a minimum learning rate of 0.00006 is reached. To learn the cell distance transformation and the cell neighbor distance transformation, PyTorch's *SmoothL1Loss* is used and both losses are added. During training flipping, scaling, rotation, contrast changing, blurring and noise adding augmentations are applied. The best model is selected manually.

Inference. A batch size of 1 is used which allows also people without GPUs with more than 12 GB VRAM to execute our code. Only for **Fluo-N3DL-TRIF** a batch size of 4 is used, and thus at least 24 GB VRAM and 32 GB RAM should be available. For 3D data, the *z*-slices are processed independently and stacked.

Post-Processing. The post-processing starts with a 2D/3D Gaussian smoothing of the predicted cell distance transform (σ_{cell}) and cell neighbor distance transform (σ_{neigh}). Then, the region to flood with a 2D/3D watershed is extracted out of the smoothed cell distance transform prediction using the threshold T_{mask}. To get markers, the smoothed cell neighbor distance transform prediction taken to the power of a (depending on how well the transforms are learned for a specific dataset) is subtracted from the cell prediction and thresholded with the threshold T_{marker}. Markers with an area smaller than 2 are filtered. The parameters for the processed cell tracking challenge data sets are shown in Table 1. For some datasets, markers are closed using a morphological closing operation (**Fluo-N3DH-SIM+**) or large artifacts are removed (**BF-C2DL-HSC** and **Fluo-N3DL-TRIC**). To process the large volumes of **Fluo-N3DL-TRIF**, only each second slice is processed and the volume additionally split into two parts with small overlap due to RAM restrictions. The final predictions are upsampled using a nearest-neighbor interpolation. For **Fluo-N3DH-CE**, merged cell nuclei along the axial axis are detected if their volume is bigger than $\frac{4}{3}$ times the mean object volume at that single frame. Detected merged nuclei are split, thresholding the cell prediction with a threshold of 0.75.

KIT-Sch-GE (1): TRACKING

First, for each segmented object at t = 0 a track is initialized. For datasets with tracking seeds, tracks are initialized only for segmentation masks including a seed. Based on the initial segmentation a region of interest (ROI) is defined for each track. At each time point t > 0 a displacement between the previous frame and the current frame for each ROI is estimated. Then, each ROI at time point t is adapted according to its displacement estimation. For the displacement estimation a cross correlation is used. Next, for each active track at time point t – *1*, a set of potential matching candidates is selected at time point t based on its ROI. A track is considered active if it has no successors and its last time point an object has been matched to it, t_{last}, is less than $t - t_{\text{last}} < \Delta t$ time points apart from t. At each time point t a linking between tracks and matching candidates is applied. For the object matching an adapted version of [3] is used. The algorithm has been adapted as follows. First, no merging events are considered. The appearance costs of objects are set to 0, as spurious tracks will be filtered out by the subsequent post-processing. The matching cost $c_{n,s}$ between track s and matching candidate n are adapted to $c_{n,s} = \|q_t^s - p_t^n\|_2$, where q_t^s is the estimated position of track s at time t and p_t^n is the position of the matching candidate. The estimated position q_t^s is given by $q_t^s = p_{t-1}^s + d^s$ where $d^s$is the estimated displacement of the ROI of track s between time points $t - 1$ and t. The splitting costs are computed based on the position of the track and its potential successors and their size. The number of nodes within the constructed graph is reduced, as each active track is only linked to segmentation mask overlapping with its ROI. Moreover, all active tracks are added to the graph and not only objects between successive time steps. This procedure helps to account for single segmentation errors as tracks can be relinked. For tracks with missing segmentation masks the position is linearly interpolated between t and t_{last} and masks are filled at the missing time steps. For the end-to-end tracking datasets, each non-matched object at time point t is initialized as a new track.

KIT-Sch-GE (1): POST-PROCESSING

Tracks of length one without any predecessor and successors are removed. Furthermore, frames without any tracked object are replaced by the tracking result of the temporally closest frame.

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. Stegmaier J. New methods to improve large-scale microscopy image analysis with prior knowledge and uncertainty. PhD thesis, Karlsruher Institut für Technologie (KIT), 2017.
3. Padfield D, Rittscher J, Roysam B. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis* **15**, 650-668 (2011).