Embedding-based Instance Segmentation in Microscopy

Manan Lalit[1,2], Pavel Tomancak[1,2,3], Florian Jug[1,2,4]

lalit@mpi-cbg.de, tomancak@mpi-cbg.de, jug@mpi-cbg.de

[1] Max Planck Institute of Molecular Cell Biology and Genetics,
[2] Center for Systems Biology Dresden,
[3] IT Innovations, VSB -Technical University of Ostrava,
[4] Fondazione Human Technopole, Milano

Abstract

Automatic detection and segmentation of objects in 2D and 3D microscopy data is important for countless biomedical applications. In the natural image domain, spatial embedding-based instance segmentation methods are known to yield high-quality results, but their utility for segmenting microscopy data is currently little researched. Here we introduce EmbedSeg, an embedding-based instance segmentation method which outperforms existing state-of-the-art baselines on 2D as well as 3D microscopy datasets. Additionally, we show that EmbedSeg has a GPU memory footprint small enough to train even on laptop GPUs, making it accessible to virtually everyone. Finally, we introduce four new 3D microscopy datasets, which we make publicly available alongside ground truth training labels. Our open-source implementation is available at https://github.com/juglab/EmbedSeg

Method

Following Neven et al 's work [1], we train a neural network to predict 5 outputs in 2D (and 7 outputs in 3D) per pixel - these are offsets in x, y (and z) of each pixel to the object center, margin bandwidths in x, y (and z) of each pixel and a seediness score. Different from Neven et al, where the authors considered the centroid to be the object center, we choose the medoid since it has the desirable property of always lying within the object. During training, we directly optimize for the intersection-over-union (IoU) metric using the Lovasz-Softmax Loss [2].

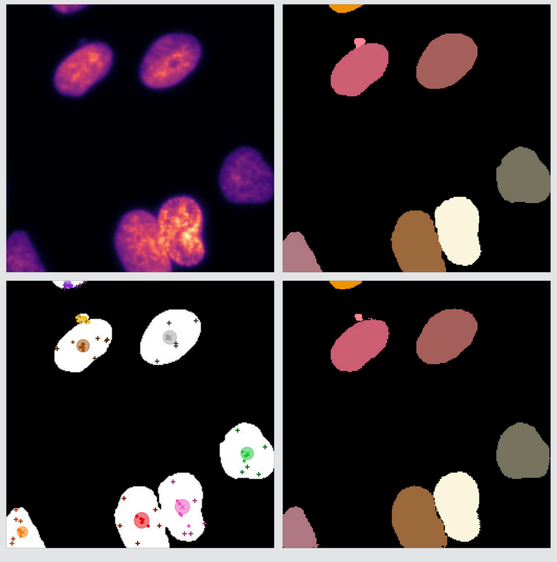

Fig. 1. Visualization of the training procedure of EMBEDSEG. An exemplary input image (top-left) and the corresponding target ground-truth instance mask (top-right) are shown. The foreground pixel locations for each object are concatenated with the corresponding offset predictions to give the resultant pixel spatial embeddings. Next, by querying which pixel embeddings give a likelihood greater than 0.5, we can back-track and identify the source pixels and thus in turn calculate the final predicted instance mask (bottom-right). In the bottom left image we show: ground truth instances as a binary mask (white regions), the embedding location and the instance clustering bandwidth (thresholded at a likelihood of 0.5) as a larger, semi-transparent ellipse, and the learnt spatial embedding locations (smaller dots inside ellipses) of 5 randomly chosen foreground pixels per predicted instance (colored plus signs).

Qualitative Results - 2D

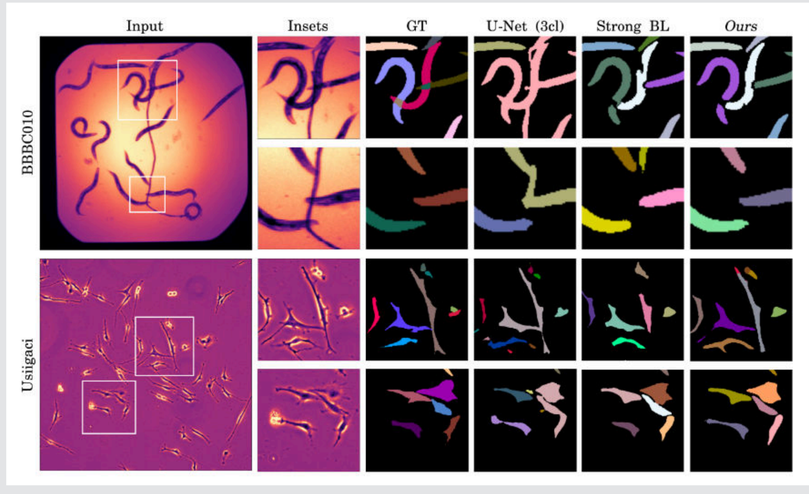

Fig. 2. Qualitative results. EMBEDSEG and two baselines compared on representative images of the BBBC010 and Usiigaci datasets. Columns show: full input image, zoomed insets, ground truth labels (GT), and instance segmentation results by the 3-class U-Net baseline, the best performing competing baseline, and EMBEDSEG. Note that each segmented instance is shown in a unique random color.

Quantitative Results - 2D

	GPU_{GB}	$AP_{0.50}$	$AP_{0.55}$	$AP_{0.60}$	$AP_{0.65}$	$AP_{0.70}$	$AP_{0.75}$	$AP_{0.80}$	$AP_{0.85}$	$AP_{0.90}$
BBBC010										
3-Class Unet	5.6	0.521	0.466	0.451	0.440	0.427	0.407	0.377	0.332	0.243
Cellpose (public)		0.225	0.204	0.184	0.155	0.097	0.043	0.013	0.002	0.000
Harmonic Emb.		0.900					0.723			
PatchPerPix		0.930		0.905		0.879		**0.792**		**0.386**
Neven *et al.*	<1	0.953	0.941	0.927	0.904	0.878	0.830	0.731	0.563	0.297
EMBEDSEG (*Ours*)	<1	**0.965**	**0.954**	**0.934**	**0.917**	**0.896**	**0.854**	0.762	**0.596**	0.326
Usiigaci										
3-Class Unet	5.6	0.245	0.188	0.133	0.090	0.049	0.016	0.008	0.000	0.000
Cellpose (public)		0.291	0.237	0.169	0.128	0.066	0.031	0.010	0.000	0.000
Cellpose (*Usiigaci*)	3.6	**0.704**	0.600	0.499	0.370	0.258	0.138	0.040	0.005	0.000
Mask R-CNN	6.9	0.583	0.520	0.439	0.365	0.235	0.130	**0.045**	**0.008**	0.000
StarDist	6.9	0.510	0.427	0.337	0.235	0.143	0.076	0.019	0.002	0.000
Neven *et al.*	**2.9**	0.648	0.570	0.463	0.343	0.233	0.115	0.035	0.004	0.000
EMBEDSEG (*Ours*)	**2.9**	**0.704**	**0.643**	**0.535**	**0.414**	**0.273**	**0.140**	0.044	0.005	0.000
DSB										
3-Class Unet	5.6	0.806	0.775	0.743	0.701	0.654	0.578	0.491	0.374	0.226
Cellpose (public)		0.868	0.852	0.829	0.802	0.755	0.676	0.563	0.418	0.234
Cellpose (*DSB*)	3.6	0.853	0.826	0.812	0.792	**0.768**	**0.716**	**0.645**	**0.536**	**0.402**
Mask R-CNN	6.9	0.832	0.805	0.773	0.730	0.684	0.597	0.489	0.353	0.189
PatchPerPix		0.868		0.827		0.755		0.635		0.379
StarDist	6.9	0.864	0.836	0.804	0.755	0.685	0.586	0.450	0.287	0.119
Neven *et al.*	**1.3**	0.873	0.852	0.830	0.799	0.762	0.704	0.623	0.511	0.373
EMBEDSEG (*Ours*)	**1.3**	**0.876**	**0.858**	**0.834**	**0.806**	**0.768**	0.715	**0.645**	0.530	0.399

Table 1. Quantitative evaluation on three 2D datasets. For each dataset, we compare results of multiple baselines (rows) to results obtained with our proposed pipeline (EMBEDSEG) highlighted in gray. First results column shows the required GPU-memory (training) of the respective method. The remaining columns show the Mean Average Precision (AP_{dsb}) for selected IoU thresholds. Best and second best performing methods per column are indicated in bold and underlined, respectively.

Qualitative Results - 3D

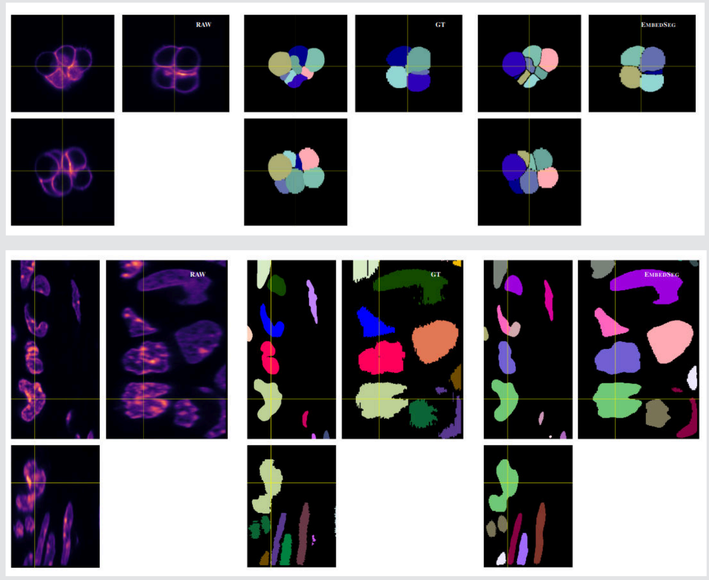

Fig. 3. Qualitative results of EMBEDSEG on the Mouse-Organoid-Cells-CBG (top) and Mouse-Skull-Nuclei-CBG (bottom) datasets. Columns show orthogonal XY, YZ and XZ slices of one representative input image, ground truth labels (GT), and our instance segmentation results using EMBEDSEG, respectively. Note that each segmented instance is shown in a random but unique color.

Training and Inference using napari [3] plugin

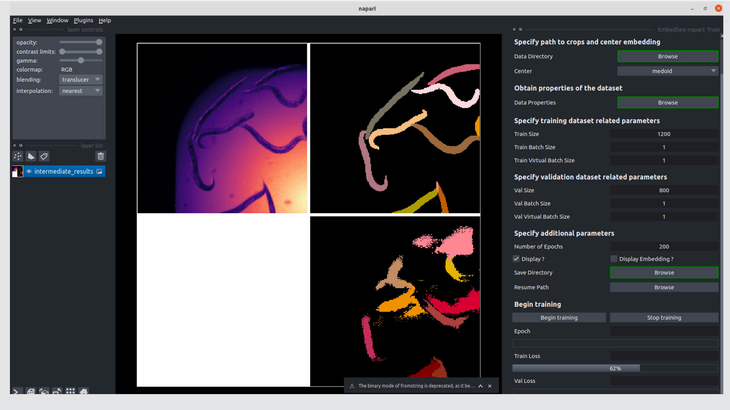

Fig. 4. We integrated EMBEDSEG as a napari plugin at https://github.com/juglab/EmbedSeg-napari Owing to its low GPU memory requirement (which we enable by activating virtual batching), we hope that EMBEDSEG can be used to train and predict on many commonly available laptops.

MIDL Conference, Lübeck, Germany, 2021

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

References

[1] Davy Neven, Bert de Brabandere, Marc Proesmans, and Luc Van Gool. **Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth.** In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019
[2] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. **The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks.** In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[3] napari contributors (2019). **napari: a multi-dimensional image viewer for python.** doi:10.5281/zenodo.3555620

Preprint

Github