Instance Segmentation of Dense and Overlapping Objects via Layering

Authors: Long Chen [1], Yuli Wu [1] and Dorit Merhof [2]

1. Institute of Imaging & Computer Vision RWTH Aachen University Aachen, Germany
2 Faculty of Informatics and Data Science, University of Regensburg, Germany

Introduction:

We propose to solve instance segmentation problem by distributing crowded, even overlapping objects into different layers. The model learns to only group spatially separated objects into the same layer (layering).

In comparison to previous methods, our approach is not affected by complex object shapes or object overlaps. With minimal post-processing, our method yields very competitive results

Approaches:

The model is trained in 2 phases: layering training and overlap completion with the following loss:

$$L = [L_{foreground}]_S^{1,2} + [L_{layering}]_{S_{fn}}^{1,2} + [L_{overlap}]_{S_f}^{2}$$

, where the subscript of [·] denotes on which area a loss term is computed: S, S_f and S_{fn} represent the whole image, the foreground and the foreground without object overlap. The superscript indicates in which training phase the loss term is included.

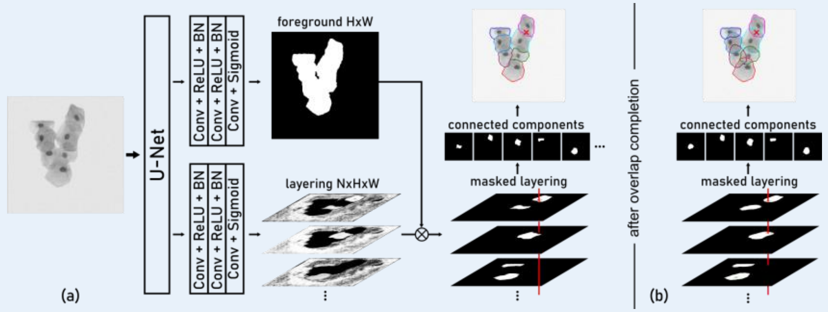
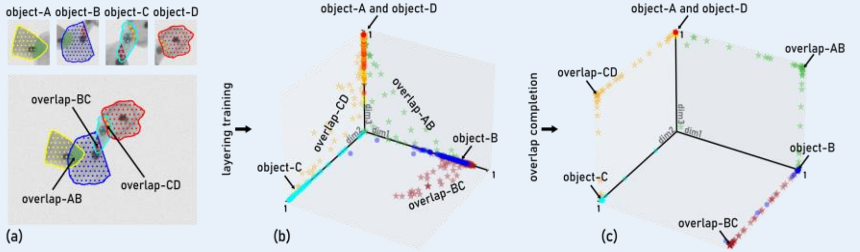


Figure 1: Overview of the model and training phases.

- Layering training: the model learns to assign foreground pixels to one of the layers, maintaining the restriction that neighboring objects should be located in different layers.

$$L_{layering} = L_{attr} + L_{rep} + L_{sparse}$$

L_{attr}: draw pixels of the same object together
L_{rep}: push neighboring objects into orthogonal space
L_{sparse}: imposes preferences for one-hot vector



- Overlap completion:
  - generate N binary masks by ordering each object mask into one of the N layers, based on the layering results of the object's overlap-free part
  - apply the *Dice*-like loss term $L_{overlap}$ with the generated masks as pseudo ground truth

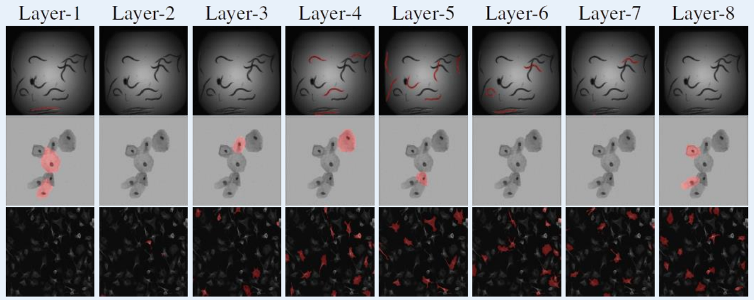
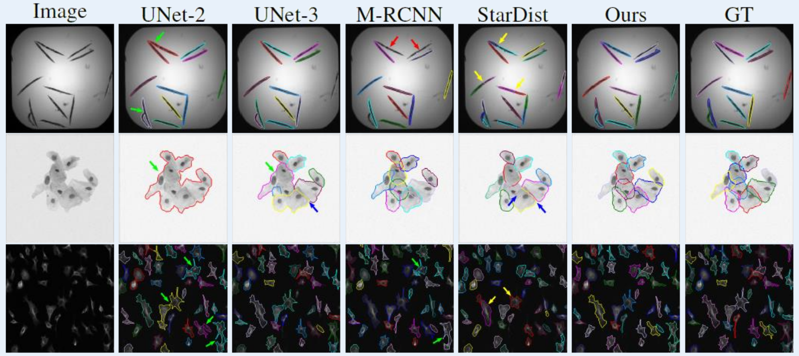


Figure 2: Activation map of each layer. Spatially adjacent objects are successfully distributed to different layers.

Results:

- Qualitative results: a few typical errors is marked (red: false suppression, green: merged objects, yellow: inaccurate shape, blue: incapability to handle overlap)



- Quantitative results: Average precisions (AP) under different IoU thresholds, mean average precision, and the Aggregated Jaccard Index (AJI) are reported

Data and Methods		$AP_{0.5}$	$AP_{0.6}$	$AP_{0.7}$	$AP_{0.8}$	$AP_{0.9}$	$meanAP$	AJI
BBBC	UNet-2	.5455	.4645	.4497	.4212	.2355	.4233	.5327
	UNet-3	.8863	.8197	.7412	**.5795**	**.2717**	**.6597**	**.7786**
	StarDist	.3098	.1410	.0372	.0011	.000	.0978	.4499
	MRCNN	**.8953**	**.8629**	**.8111**	.5305	.0382	.6276	.7580
	Ours	**.9357**	**.9188**	**.8648**	**.7606**	**.2904**	**.7541**	**.8442**
OCC	UNet-2	.1548	.1303	.1129	.1055	.1048	.1217	.2058
	UNet-3	.7010	.6071	.5097	.3767	.1950	.4779	.5802
	StarDist	.6556	.5566	.4346	.2970	.1547	.4197	.6927
	MRCNN	**.9277**	**.9181**	**.8870**	**.8117**	**.5564**	**.8202**	**.8412**
	Ours	**.9230**	**.8768**	**.8007**	**.6788**	**.4349**	**.7429**	**.8353**
CCDB	UNet-2	.3698	.3360	.3049	.2763	.2228	.3020	.1185
	UNet-3	.7307	**.6774**	**.6210**	.5153	.2838	**.5656**	**.7148**
	StarDist	**.7428**	.6532	.4958	.2685	.0326	.4386	.6903
	MRCNN	.6248	.5691	4888	3476	.0763	.4213	.5842
	Ours	**.7968**	**.7467**	**.6767**	**.4889**	**.2230**	**.5864**	**.7601**

(for detailed expression of the loss terms and supplementary experimental results, refer to the paper)