Unsupervised Learning of Object-Centric Embeddings for Cell Instance Segmentation in Microscopy Images

Steffen Wolf[1], Manan Lalit[2], Henry Westmacott[1], Katie McDole[1*], Jan Funke[2†]

[1]MRC Laboratory of Molecular Biology, [2]HHMI Janelia Research Campus

Abstract

Segmentation of objects in microscopy images is required for many biomedical applications. We introduce object-centric embeddings (OCEs), which embed image patches such that the spatial offsets between patches cropped from the same object are preserved. Those learnt embeddings can be used to delineate individual objects and thus obtain instance segmentations. Here, we show theoretically that, under assumptions commonly found in microscopy images, OCEs can be learnt through a self-supervised task that predicts the spatial offset between image patches. Together, this forms an unsupervised cell instance segmentation method which we evaluate on nine diverse large-scale microscopy datasets. Segmentations obtained with our method lead to substantially improved results, compared to state-of-the-art baselines on six out of nine datasets, and perform on par on the remaining three datasets. If ground-truth annotations are available, our method serves as an excellent starting point for supervised training, reducing the required amount of ground-truth needed by one order of magnitude, thus substantially increasing the practical applicability of our method. Source code is available at github.com/funkelab/cellulus.

1. Introduction

Determining whether two image regions belong to the same object is a fundamental challenge in instance segmentation, albeit a simple task for humans. A plausible hypothesis is that humans learn to recognize parts as belonging to a whole by repeatedly observing them in each other's vicinity. We introduce object-centric embeddings (OCEs), which leverage this assumption for unsupervised instance segmentation. OCEs map image patches in such a way that the spatial offsets between patches cropped from the same object are preserved in embedding space. We investigate the us-

age of OCEs in the domain of microscopy imaging and introduce CELLULUS, a method that identifies and segments individual cells in microscopy images.

By relying on reasonable assumptions about microscopy images, namely that (i) the objects in these images have a similar appearance and (ii) the objects in these images are randomly distributed, we show that OCEs can be learnt in an unsupervised fashion.

Cell instance segmentation is crucial for answering important life science questions. In recent years, deep learning-based segmentation approaches [23, 12] have achieved the best performance on standard benchmarking datasets, but these approaches rely on large amounts of annotated training data. Our proposed unsupervised method CELLULUS, in contrast, circumvents the problem of acquiring these manual annotations.

With CELLULUS, we provide an approach for employing the learnt object-centric embedding locations per patch, identifying image patches that are part of the same cell and thus segmenting cell instances in a unsupervised way (see Figure 1 for a few examples). We demonstrate that this unsupervised segmentation pipeline achieves competitive results with respect to pre-trained baseline models on a diverse set of nine microscopy image datasets (see Table 1).

Additionally, instance segmentations obtained through our proposed unsupervised pipeline are excellent starting points to support supervised training when very little manually generated ground truth annotations are available. We show that we obtain comparable performance to supervised segmentation methods, after fine-tuning on one order of magnitude less data (see Figure 5).

More generally, supervised training supported by unsupervised segmentation is at least as good as purely supervised learning on all investigated datasets, demonstrating that our method dramatically reduces the amount of ground truth annotations needed, and at times not requiring any.

Reducing or eliminating the need for manual ground truth is of particular importance to biological research, as new light-microscopy methods are capable of generating terabytes of data in a single experiment. Manually annotating even small regions of such datasets can take hundreds

---

*Corresponding Author, kmcdole@mrc-lmb.cam.ac.uk.
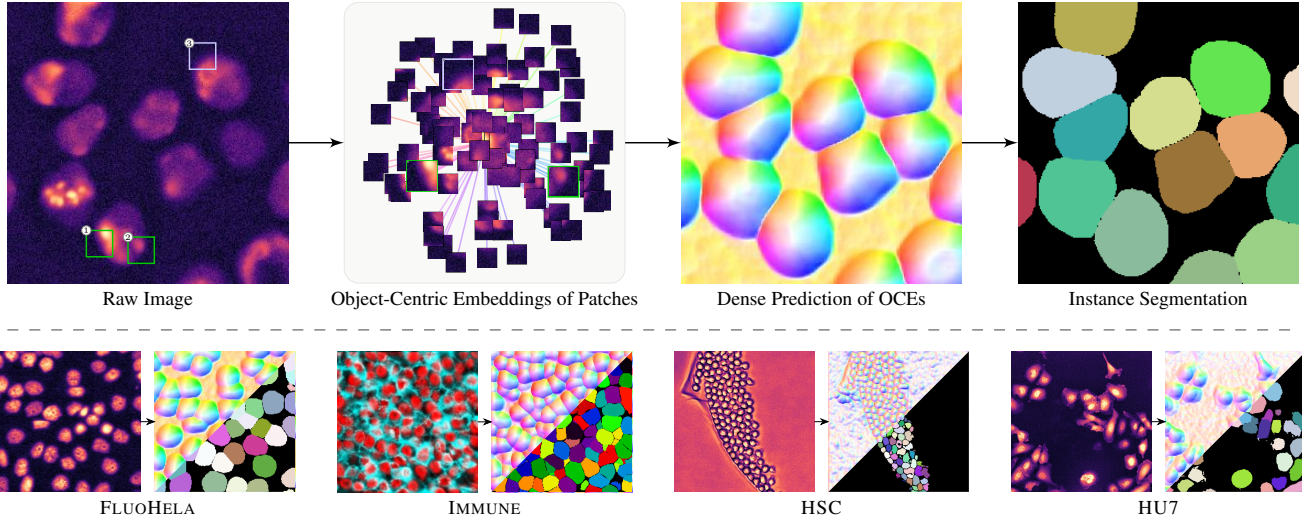
†Corresponding Author, funkej@janelia.hhmi.org.

Figure 1. Method overview and example segmentations on diverse datasets. Top row: An unsupervised learning objective gives rise to object-centric embeddings (OCEs), such that patches extracted from the same object (green boxes) maintain their relative position to each other. Predicted densely, these OCEs allow instance segmentation of cells in microscopy images, by using a post-processing step such as mean-shift clustering. Bottom row: Example raw images and dense OCEs/instance segmentations on four datasets spanning different imaging modalities, cell sizes and shapes.

or thousands of human hours. Thus, there is a tremendous need for self-supervised learning methods to help cope with the vast amount of data generated by modern microscopes. CELLULUS is available at github.com/funkelab/cellulus.

2. Related Work

Currently, machine learning and deep learning-based methods dominate the field of cell instance segmentation [26, 23, 12]. These cell segmentation methods can be categorized by their intermediate (auxiliary) representation used to derive the predicted segmentation.

STARDIST [22], for example, represents objects as star-convex polygons (*i.e.*, distances from a center point to the cell boundary along sets of equi-distant rays). On the other hand, CELLPOSE [23] encodes cells by vectors that point inwards from the boundary. The representations of STARDIST and CELLPOSE are pre-defined and tailored to the tasks of cell segmentation.

Alternatively, pixel-level representations (here referred to as *embeddings*) can be learnt from labels directly by pulling embeddings of pixels within instances together and pushing embeddings across instances apart [3]. Initially developed for natural images, this concept was further developed into a cell segmentation and tracking algorithm in the work by Payer *et al*. [18], which established the state-of-the-art on six Cell Tracking Challenge (CTC) datasets.

Recent submissions to the CTC further improved the segmentation and tracking performance. While Arbelle *et al*. [1] and Scherr *et al*. [21] relied on boundary classifica-

tion to separate densely clustered cells, Löffler *et al*. [15] used *spatial embeddings*.

Spatial embedding-based approaches learn a function which associates each pixel at location i in the raw image, to a relative spatial embedding (offset vector) r_i, such that the resulting absolute spatial embedding $e_i = i + r_i$ for all pixels belonging to an object instance point to a common point (*e.g.* the instance centroid).

Typically, the embeddings are learnt using a regression loss function, either minimizing the distance between absolute spatial embeddings of pairs of pixels i, j from the same instance $\mathcal{L}_{\text{regr}} = \sum_{i,j} \sigma(e_i - e_j)$ or equivalently by approaching the mean over the whole instance [16, 12]. Here, σ is a measure of distance, *e.g.*, $|\cdot|^2$. Recently, EMBEDSEG used spatial embeddings to establish the state-of-the-art on multiple 2D and 3D microscopy datasets [12].

We note that our learning approach has parallels with supervised learning of pixel-wise spatial embeddings. In our work, self-supervised learning leads to object-centric embeddings, which are post-processed using the mean-shift clustering algorithm, analogous to De Brabandere *et al*. [3].

Self-supervised learning methods learn representation by solving tasks that predict an intentionally hidden part of the data. Predicting the spatial arrangement of image patches provides a rich signal for learning meaningful representations for downstream tasks. Spatial tasks include solving jigsaws [17], predicting patch rotations [6], or classifying relative patch positions from a grid-like pattern [5]. More recently, contrastive learning between multiple views [8, 9, 25, 14] enabled learning of representations
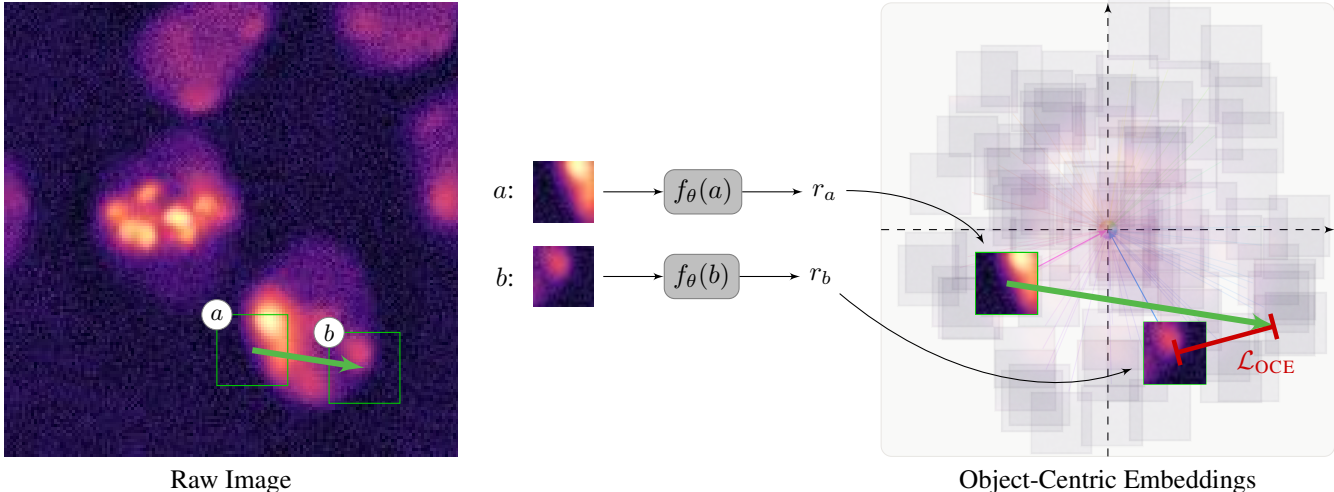
Figure 2. **Unsupervised Learning of Object-Centric Embeddings**. During learning, small image patches are randomly cropped from the raw image and embedded through a learnable function f_θ into a 2D embedding space. The objective of the loss $\mathcal{L}_{\text{OCE}}$ is to ensure that the spatial offset between pairs of patches in the raw image (green arrows) is preserved in the embedding space (see Equation 4).

that transfer well to downstream tasks. These learnt representations have been shown to reduce the required amounts of annotated data in tasks such as image classification [10] and semantic segmentation [13, 24].

2.1. Unsupervised Methods for Cell Segmentation

Recently, methods for cell instance segmentation have been proposed that do not rely on human annotation.

The unsupervised segmentation pipeline proposed by Din and Yu [4] employed a Convolutional Neural Network (CNN), which when centered on each cell nucleus is tasked to predict a binary mask for each cell. The model is trained without any ground-truth and is tasked to predict consistent masks that cover all foreground pixels. However, this method still relied on pre-trained networks for locating the nuclei using which the cell segmentations are predicted and can therefore not be considered fully unsupervised.

Completely unsupervised *instance separation* has been proposed by Wolf *et al*. [27], where inpainting networks are used to determine which image regions are independent. These independent regions are determined by a hierarchical optimization strategy that continually subdivides the image until all instances are separated. In contrast to our proposed method CELLULUS, the post-processing step of Wolf *et al*. [27] is very computationally expensive and does not provide a method for detecting background regions automatically.

Xie *et al*. [28] proposed a self-supervised method that employed two proxy tasks of estimating nuclei size and ranking count of nuclei and this enabled the model to mine instance-aware representations from raw data.

3. Method

We aim to learn an embedding of image patches that reflects the relative spatial arrangement of these patches (*i.e.* the offset between the predicted embeddings should be equal to their spatial offset), as if they were extracted from the same object (see Figure 2). We refer to the spatial offset between patches extracted from the same object as *intra-object offset* and the learnt embeddings as *object-centric embeddings* (OCEs).

3.1. Unsupervised Learning of OCEs

Under conditions that are commonly found in microscopy images, OCEs can be learnt in an unsupervised manner, *i.e.*, without the provision of segmentation ground-truth. Those conditions are:

1. Objects in the image are similar
2. Objects are randomly distributed in the image plane
3. A patch cropped from an object contains enough information to identify its position inside the object (*i.e.*, no two parts of an object look exactly identical)

Under these conditions, the *expected* offset between two image patches is proportional to the *intra-object offset* of those two patches, *i.e.*, the spatial offset between those patches if they were part of the same object.

Let a and b be two different patches found on an object (*e.g.*, the left- and the right-most patches of a cell, see Figure 3). If multiple similar objects are present in an image, there will be multiple locations $i \in \Omega$ where the patch a is visible, and distinct locations $j \in \Omega$ where the patch b is visible. Here, Ω is the set of all pixel locations and $x : \Omega \mapsto \mathcal{R}$ is the image. We will refer to the image patch at a location
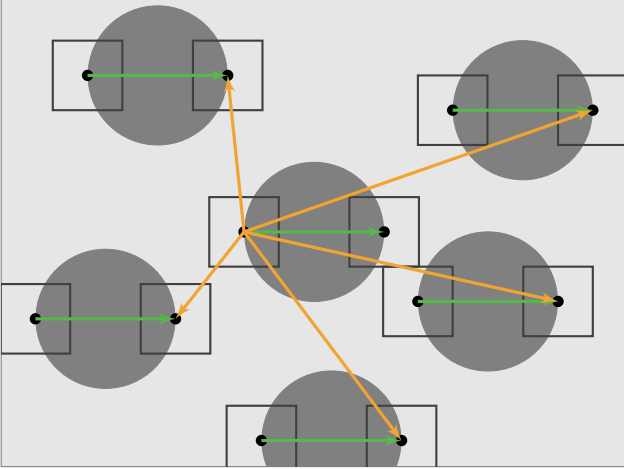
Figure 3. **Illustration of the expected offset between two example patches in an idealized image**: Black squares show all image locations where the two patches ▮ and ▮ are found. The expected offset between those patches stems from offsets observed within the *same* object (*intra-object offsets*, shown as green arrows) and offsets observed between *different* objects (*inter-object offsets*, shown as orange arrows for the center object only). Assuming a random distribution of objects in large images, the average offset between *different* objects is zero, thus the expected offset $\overrightarrow{ij}$ between the given patches is proportional to the intra-object offset.

i as $p(i)$ and denote the set of all locations that contain a given patch a as Ω_a, *i.e.*, $\Omega_a = \{i \in \Omega \mid p(i) = a\}$.

Consider the expected observed offset $\overrightarrow{ij} = j - i$ between all occurrences of patches a and b: for each object contained in the image, patches a and b are observed once with their *intra-object offset*, *i.e.*, the offset they have to each other as being part of the same object. For every pair of different objects, however, patches a and b will be observed at random offsets, following the assumption that objects in the image are randomly distributed. The key insight that allows unsupervised learning of OCEs is that the observed offsets of patches from different objects have zero mean.

Formally, the expected offset between all locations of two image patches a and b is given as

$$\mathbb{E}\left[\overrightarrow{ij}|a,b\right] \approx \frac{1}{N} \sum_{i \in \Omega_a} \sum_{j \in \Omega_b} \overrightarrow{ij}, \qquad (1)$$

where $N = |\Omega_a| \cdot |\Omega_b|$ is the number of pairs of image locations i, j, where patches a and b are observed.

This expectation can be rewritten to distinguish observed offsets from the *same* versus *different* objects. For that, let Ω_b^i denote all locations j where patch b appears and is part of the same object at location i. Similarly, let $\overline{\Omega}_b^i$ be the set of locations j where patch b appears, but is not part of the object at location i. We can now rewrite the expected

observed offset $\overrightarrow{ij}$ as

$$\mathbb{E}\left[\overrightarrow{ij}|a,b\right] \approx \frac{1}{N} \sum_{i \in \Omega_a} \left[\sum_{j \in \Omega_b^i} \overrightarrow{ij} + \sum_{j \in \overline{\Omega}_b^i} \overrightarrow{ij} \right] \qquad (2)$$

$$= \underbrace{\frac{1}{N_{\mathrm{s}}} \sum_{i \in \Omega_a} \sum_{j \in \Omega_b^a} \overrightarrow{ij}}_{\text{intra-object offset}} + \underbrace{\frac{1}{N_{\mathrm{d}}} \sum_{i \in \Omega_a} \sum_{j \in \overline{\Omega}_b^a} \overrightarrow{ij}}_{\approx 0}, \quad (3)$$

where N_{s} and N_{d} denote the number of times that patches a and b are observed in the same object and different objects, respectively.

The first term in Equation 3 is, by definition, the intra-object offset, *i.e.*, the quantity we aim to infer. The second term is the expected offset between patches a and b if both are part of different objects. Under the assumption that multiple similar objects are randomly distributed in the image, this expectation is zero: observing patch a relative to patch b with offset $\overrightarrow{ij}$ is just as likely as observing them at the inverse offset $\overrightarrow{ji}$. Without any supervision, the constants N_{s} and N_{d} are not known. The expected offset, calculated as in Equation 1, is thus proportional to the sought after intra-object offset.

In conclusion, the expected offset given any two patches approximates the offset between the patches extracted from the same object. We can leverage this property to devise a loss function that minimizes the differences between the spatial and embedding offsets between pairs of patches, and thus learn an object-centric embedding in an unsupervised fashion.

Let $f_\theta : \mathcal{P} \mapsto \mathbb{R}^2$ be a parameterized embedding function, mapping from the set of all image patches $\mathcal{P}$ to a 2D embedding space. We denote a patch located at i as $p(i)$ and its embedding as $r(i) = f_\theta(p(i))$. We propose the following unsupervised loss, minimizing the difference between $d(i,j) = i - j$ and $\hat{d}(i,j) = r(i) - r(j)$ for pairs of patches:

$$\mathcal{L}_{\mathrm{OCE}} = \sum_{i,j \in \Omega} \sigma\left(d(i,j) - \hat{d}(i,j)\right), \qquad (4)$$

where σ is a measure of distance, *e.g.*, $|\cdot|_2$ (we will discuss our choice of σ below).

3.2. Loss Implementation

In practice, the embedding function will be implemented as a Convolutional Neural Network (CNN) and its weights can be updated using stochastic gradient descent. In this setting, strong gradient contributions resulting from pairs of patches of different objects can be problematic due to their high variance, even if they have zero mean. To address this, we dampen the effect of large distances in our loss function by using a sigmoid distance function, *i.e.*,
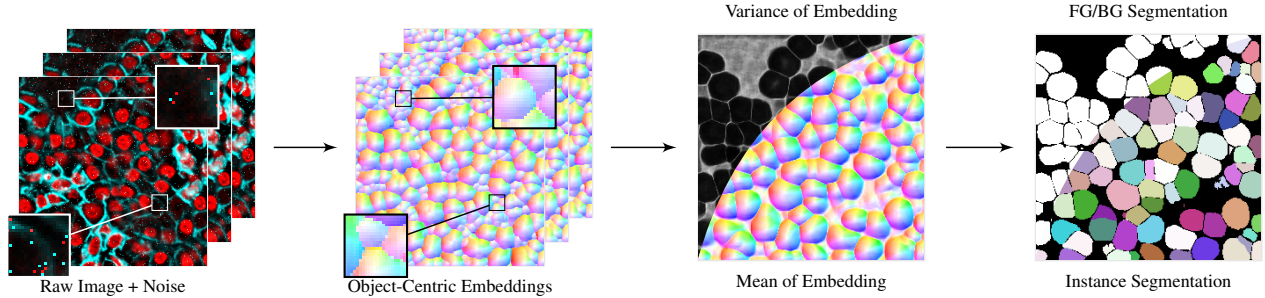
Figure 4. **Overview of the inference pipeline**. Input image to the trained object-centric embedding (OCE) network is augmented repeatedly with salt and pepper noise, producing several noisy instances of the raw image (first column). OCEs are predicted densely for each noisy instance of the input raw image (second column). Next, the pixel-wise mean and variance of the predicted OCEs is calculated (third column). Images locations with high variance are treated as the background. The remaining foreground region is clustered into individual object instances using mean-shift clustering (fourth column).

$\sigma(\delta) = \left(1 + \exp\left(-\frac{\|\delta\|_2^2}{\tau}\right)\right)^{-1}$, where τ is a hyperparameter controlling the rate of damping.

Furthermore, we limit the sampling of pairs of patches to have a maximal distance κ and add an L2 regularization term to obtain our final unsupervised loss function as

$$\mathcal{L} = \sum_{i,j \in P} \sigma\left(d(i,j) - \hat{d}(i,j)\right) + \lambda_{\text{reg}} \|r(i)\|_2, \quad (5)$$

where $P \subset \{i, j \in \Omega \mid |i - j|_2 \leq \kappa\}$. For more details, see Appendix A.

3.3. Instance Segmentation from OCEs

An instance segmentation can be obtained from OCEs by firstly segmenting foreground vs. background, followed by partitioning the foreground into individual instances.

To address the background identification, we exploit the sensitivity of the OCEs to noise in background: We observe that certain noise patterns in the background (*e.g.*, single bright pixels) become the center point of locally consistent embeddings, thus creating spurious objects (see Figure 4, first column for an example). To identify background, we repeatedly introduce artificial noise to the raw image and measure the variance of the predicted embeddings (we found salt-and-pepper noise to be effective). We find that the distribution of the variance of these embeddings over image locations is bi-modal, such that a parameter-free thresholding method like Otsu's is sufficient to separate foreground from background.

After identifying the background, we segment individual instances in the foreground through a mean-shift clustering on the dense OCE predictions [2, 19] (see Figure 4).

4. Experiments

Used Datasets. We test our method CELLULUS on nine publicly available datasets for which dense ground truth an-

notations are available. The datasets were chosen to represent a diverse set of image modalities, cell/tissue types, and imaging platforms.

TissueNet [7] is the largest of the analyzed datasets, with 1.3 million annotated cells. It covers six imaging platforms and includes histologically normal and diseased tissue of humans, mice, and macaques. The included tissue types (IMMUNE, LUNG, PANCREAS, SKIN cells) vary widely in cell appearance and density. Therefore, we add evaluations where we restrict the dataset to the four individual tissue types. For reference, constructing *TissueNet* required $> 4,000$ hours of human annotation time.

The nuclei and whole cell are labeled in *TissueNet* and both of these image channels were used during training and inference. For evaluation purposes during inference, the predicted instance segmentations are compared against the ground truth labels for the whole cell image channel.

Cell Tracking Challenge (CTC) [26] provides diverse 2D and 3D datasets [1]. We select five 2D datasets with distinct cell appearances: HSC, HU7, SIMULATED, FLUOHELA and PSC.

Each dataset comes with two sets of image sequences: 1 and 2. We used set 1 for training, while set 2 is held out for evaluation. Images in the *CTC* datasets contain only one channel.

Segmentation Metrics. We use two widely used cell segmentation scores: (i) SEG score (used by CTC [26]) matches every ground truth object to a predicted instance segmentation and measures the average intersection over union (IOU) of all matches. (ii) F1 score (used by Greenwald *et al.* [7]) matches all predictions and ground truth objects with an IOU greater than or equal to a fixed threshold (0.5 unless specified) and reports the F1 measure of successfully found matches.

---

[1] http://celltrackingchallenge.net/2d-datasets/

Table 1. **Quantitative results when no annotations are available** (*fully unsupervised setting*).
The pretrained models of STARDIST [22] and CELLPOSE [23] are compared with CELLULUS on nine diverse microscopy image datasets. Two instance segmentation metrics F1 and SEG are evaluated by comparing the quality of predicted instance segmentation with the ground truth instance segmentation. Best performing method on each dataset is shown in bold. The last row TISSUENET (all) shows a weighted average (weights proportional to the number of images) of results for IMMUNE, LUNG, PANCREAS and SKIN.

	CELLPOSE		STARDIST		CELLULUS	
	F1	SEG	F1	SEG	F1	SEG
HSC	0.00	0.00	**0.09**	0.14	0.06	**0.42**
HU7	0.40	0.27	0.03	0.02	**0.75**	**0.55**
SIMULATED	0.49	0.34	0.23	0.35	**0.83**	**0.65**
FLUOHELA	0.36	0.65	**0.38**	**0.79**	0.34	0.70
PSC	**0.76**	**0.58**	0.64	0.47	0.64	0.51
IMMUNE	0.44	0.21	0.66	0.41	**0.69**	**0.57**
LUNG	0.76	0.53	**0.81**	**0.59**	0.51	0.51
PANCREAS	0.56	0.36	0.58	0.36	**0.67**	**0.49**
SKIN	0.48	0.26	0.39	0.24	**0.60**	**0.46**
TISSUENET (all)	0.55	0.32	0.59	0.38	**0.64**	**0.52**

4.1. Unsupervised Segmentation

We compare CELLULUS against two state-of-the-art pretrained segmentation models that are widely used across datasets. We investigate the segmentation performance under the condition that no ground truth annotations are available.

Baseline Methods. STARDIST [22] is a widely used cell/nucleus segmentation method. It predicts, for each pixel, the distances to the boundary in a predefined set of directions. CELLPOSE [23] uses a supervised network to predict spatial embeddings and clusters pixels together using a diffusion-based aggregation method.

Segmentation Performance. For each dataset, we train an object-centric embedding network. Raw images are intensity-normalized (1 percentile intensity is mapped to 0 while 99.8 percentile intensity is mapped to 1) and input to the network to produce dense object-centric embeddings. During inference, these object-centric embeddings are processed to obtain instance segmentations and the F1 and SEG scores are computed with respect to the ground truth masks for the set of images held-out for evaluation purposes (see Table 1). An overview of the datasets and the predicted embeddings and instance segmentations is shown in Figure 7.

Our method outperforms both baselines on real-world datasets HU7, HSC, SIMULATED, IMMUNE, PANCREAS and SKIN (according to SEG score). On the SIMULATED dataset, our method performs exceptionally well (see Ta-

Table 2. **Performance of CELLULUS across a range of scale factors for two datasets.**
Two instance segmentation metrics F1 and SEG are evaluated by comparing the quality of predicted instance segmentation obtained using CELLULUS at different scale factors, with the ground truth labels at scale factor = 1.0. Scale factor is inversely related to the employed patch size.

Scale Factor	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
IMMUNE											
F1	0.12	0.33	0.48	0.44	0.64	0.69	0.69	0.66	0.67	0.59	0.58
SEG	0.20	0.28	0.38	0.36	0.48	0.57	0.55	0.57	0.57	0.56	0.54
LUNG											
F1	0.17	0.23	0.32	0.52	0.48	0.51	0.36	0.46	0.37	0.27	0.28
SEG	0.27	0.25	0.35	0.49	0.44	0.51	0.40	0.52	0.48	0.38	0.44

ble 1 and Appendix D). To highlight the success and failure modes of our method, we measure the F1 score per image and report the [0th, 25th, 50th, 75th, and 100th] percentile images for different tissue types in Figure 6.

We find that our method can compensate for some variations in object sizes. Compare, for example, the small cells in the 75th percentile image of the IMMUNE dataset with more voluminous cells in the 100th percentile (see Figure 6). However, we also observe that larger outlier objects (*e.g.*, the 0th percentile SKIN image) lead to structural under-segmentation.

Background Detection Performance. We observe that our background detection generally matches the ground truth in the datasets HU7, FLUOHELA, TISSUENET, PSC and SIMULATED, where no additional structure in the background is visible. When objects are exceptionally dim, their embeddings may vary with the added noise, which leads to them being treated as background (e.g. see Figure 6, 25th percentile in the SKIN dataset).

The HSC dataset is exceptionally challenging, with a visible culture plate in the background adding additional structure to the image. This leads to additional segments in our method (see Figure 7). All studied methods struggle to predict these segments accurately, with an F1 score below 0.1. Remarkably, our method receives a high SEG score compared to the other methods. Further analysis of this dataset, including an evaluation of segmentation metrics for all matching thresholds, can be found in Appendix C.

Scale Informs All Parameter Choices. The size of the selected image patches determines what object sizes can be detected. Patches should be smaller than individual objects but still contain meaningful features. To keep all network parameters and training setup constant, we resize the training data. Specifically, datasets HU7, PSC and SIMULATED are re-scaled by a factor of [0.5,2., $\frac{2}{3}$], respectively. All other datasets were analyzed at their native resolution.

We also explore the performance of CELLULUS across a range of scale factors for two datasets IMMUNE and LUNG (see Table 2). Note that predictions produced for any scale
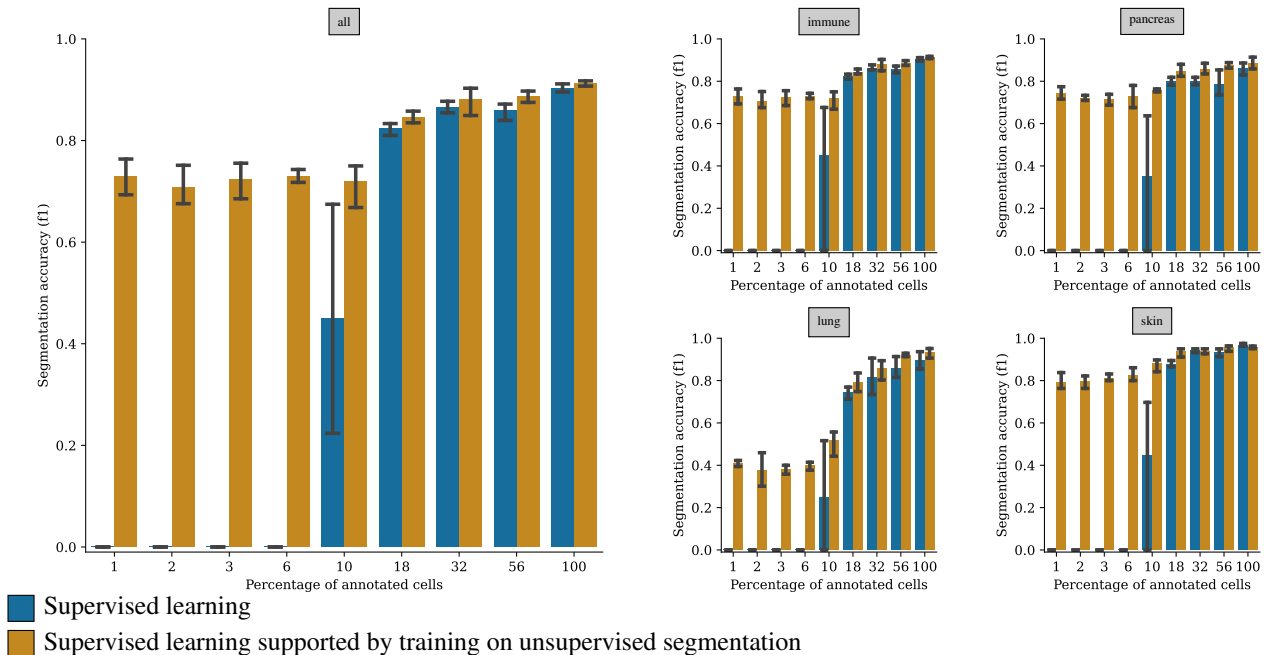
Figure 5. **Supervised cell segmentation performance for varying amounts of annotations**. We compare a classical supervised learning approach (blue) trained using only manual annotations, against using a mixture of manual annotations and pseudo-ground truth derived from our unsupervised OCEs (orange), on the four tissue types (IMMUNE, PANCREAS, LUNG and SKIN) in the TISSUENET dataset [7]. The results for ALL (left) are obtained by averaging the results obtained individually on the four datasets (right).

factor are compared against ground truth labels at scale factor = 1.0 to obtain the F1 and SEG scores.

Implementation. For learning the object-centric embeddings, we use a U-Net architecture with a limited field of view of 16×16 (single 2x down-sampling layer, ReLU activation). For more details, see Appendix A.

After the training, a scale-appropriate bandwidth of mean-shift clustering has to be chosen. We use the implementation of mean-shift clustering provided by *scikit-learn* [19] and perform a line search to determine the optimal value.

When instances are tightly packed, our segmentation matches closely with the ground truth without post-processing. However, when instances are surrounded by background, we find that patches close to the object borders get mapped to the object center. Therefore, we shrink our objects to correct for this halo (see Appendix D). We pick the optimal shrinkage distance between 0 and 6 pixels for all datasets and report the best score.

4.2. Supporting Supervised Learning

In the following experiments, we investigate how our unsupervised segmentation can be used to increase model performance when only a few objects are annotated.

Supervised Training Setup. For the supervised training setup, we build two sparsely annotated supervised datasets,

which we call the *sparse* and *pseudo dataset* by randomly sampling ground truth objects as a fixed percentage of annotated cells. (1) The *sparse dataset* contains only the annotated samples. (2) The *pseudo dataset* uses our predicted segmentations as a starting point (pseudo ground truth) and utilizes the same sampled annotations to correct our predictions. We mask all our predicted objects that overlap with the annotations and use the annotations instead. We include annotations of background pixels close to the labeled object (< 30 pixels).

We use these datasets to train a U-Net using a supervised STARDIST training loss $\mathcal{L}_{\text{STARDIST}}$ [22]. Mini-batches contain half of the images from the *sparse dataset* and the other half from the *pseudo dataset*. The STARDIST loss is computed on each respective half ($\mathcal{L}_{\text{STARDIST}}^{\text{sparse}}$ and $\mathcal{L}_{\text{STARDIST}}^{\text{pseudo}}$). The total loss $\mathcal{L}_{\text{STARDIST}} = (1-\alpha)\mathcal{L}_{\text{STARDIST}}^{\text{sparse}} + \alpha\mathcal{L}_{\text{STARDIST}}^{\text{pseudo}}$ is a linear combination, where $\alpha = 0$ corresponds to classical supervised training. For further details, see Appendix B.

We train STARDIST models with varying amounts of annotations and compare the performance of models trained only on annotated images ($\alpha = 0$, blue in Figure 5) with those supported by our segmentation ($\alpha = 0.5$, orange in Figure 5). Each experiment is repeated 3 times with different annotation samples. We evaluate the trained networks on the full TISSUENET dataset as well as subsets of tissue types IMMUNE, PANCREAS, LUNG and SKIN. All mea-
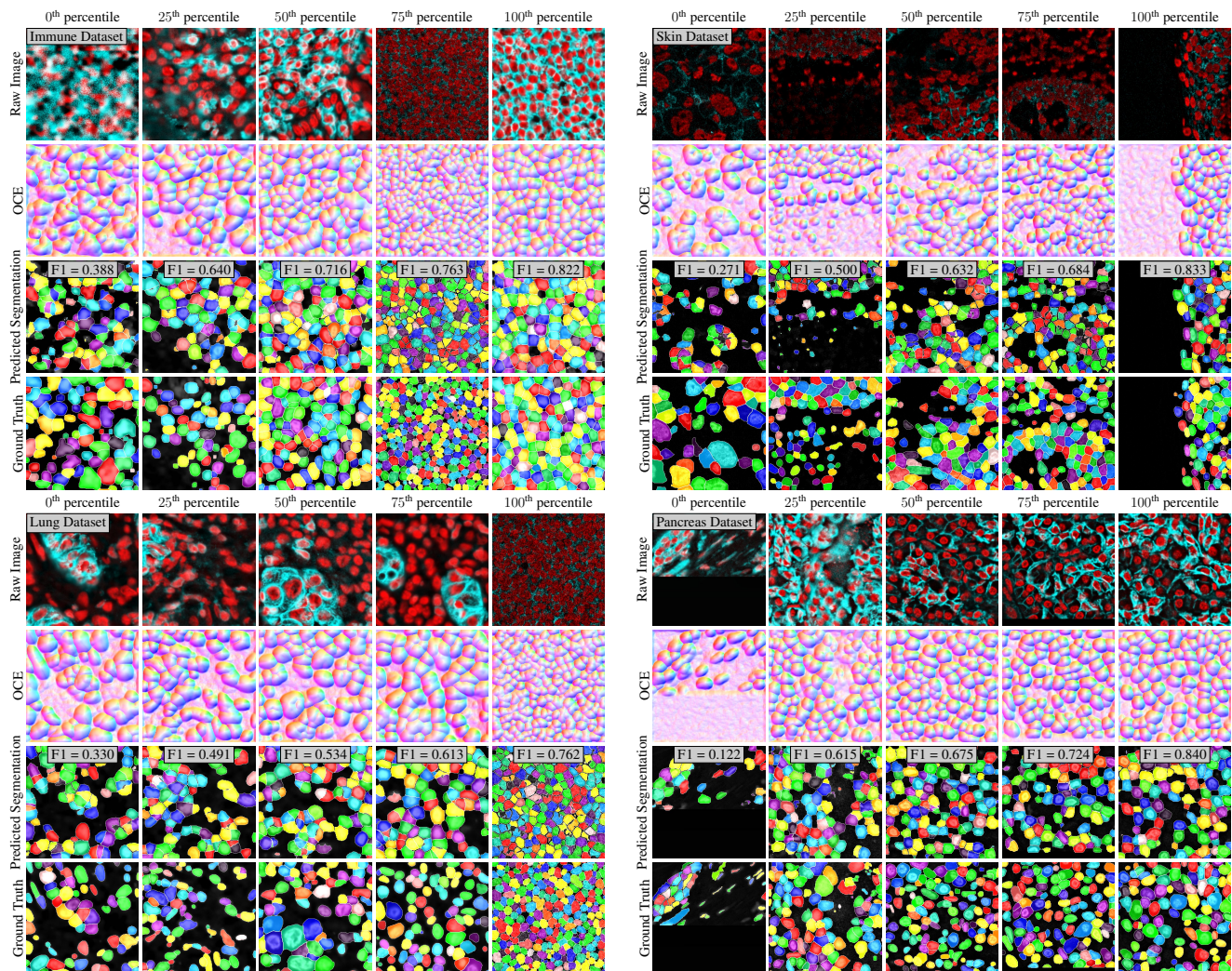
Figure 6. **Predicted OCEs and segmentations on the TISSUENET dataset with tissue types IMMUNE (top-left), SKIN (top-right), LUNG (bottom-left) and PANCREAS (bottom-right)**. The F1-score is evaluated for each individual image and the 0th, 25th, 50th, 75th and 100th percentile images and their respective F1 scores are reported in each column. Rows (from the top) show the Raw Images, Dense Prediction of OCEs, the Predicted Instance Segmentation and the Ground Truth Instance Segmentation available for evaluation purposes.

sured performances and their standard deviations are visualized in Figure 5.

Supported supervision makes a significant improvement at 1%. When 1% of cell annotations from the TISSUENET dataset are used, $F1 = 0.75 \pm 0.03$ is obtained which is significantly better than the performance of the purely unsupervised segmentation at $F1 = 0.64$ (see Table 1). This effect could be due to the biases of the STARDIST representation, which might help to refine the unsupervised segmentation.

We additionally perform a training experiment using 0% ground truth annotations (*i.e.* $\mathcal{L}_{\text{STARDIST}} = \mathcal{L}_{\text{STARDIST}}^{\text{pseudo}}$) and notice no improvement ($F1 = 0.63$). In conclusion, the combination of minimal annotations and the supporting pseudo ground truth significantly help.

Supported supervision substantially outperforms purely supervised training. Our proposed supported supervision method can be used as a replacement for training only on annotations without a performance compromise across all annotation levels. Notably, at annotation levels $\leq 10\%$ our method outperforms the baseline substantially.

5. Discussion

We believe that this work offers a feasible way to accelerate the analysis of microscopy image datasets of cells. As our experiments on nine large cell segmentation datasets demonstrate (see Table 1), a surprisingly good segmentation can often be achieved in a completely unsupervised fashion. Depending on the biological question at hand, those results might already be sufficient for downstream analy-
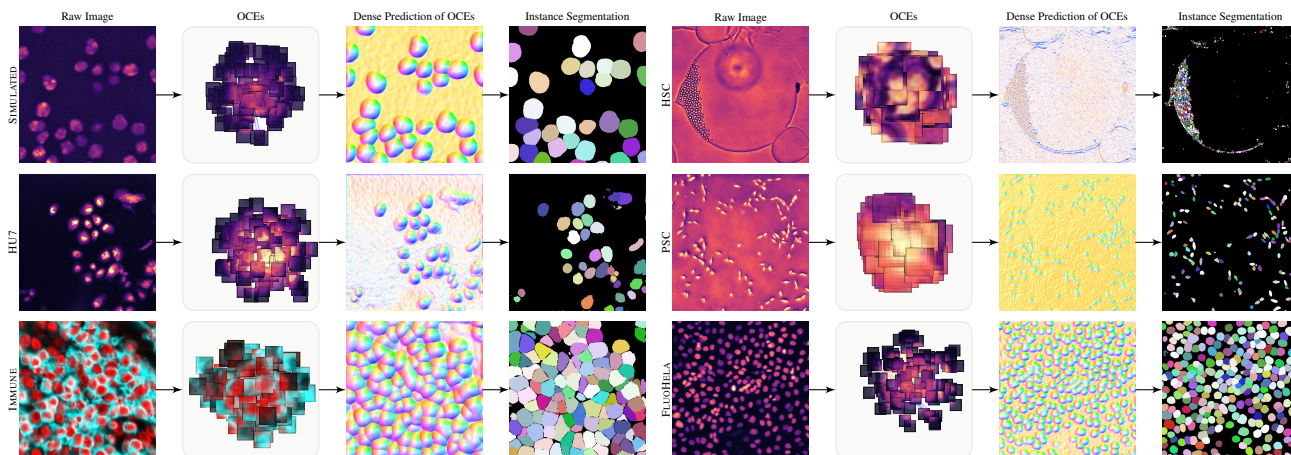
Figure 7. **Qualitative results on a diversity of microscopy image datasets**. Sample raw images from six datasets (SIMULATED, HU7, IMMUNE, HSC, PSC, FLUOHELA) are shown in the first and fifth columns. OCEs are predicted from patches extracted from these raw images and the spatial arrangement of these OCEs are shown in the second and sixth columns. During inference, OCEs are predicted densely for the input raw image (third and seventh columns). The final cell instance segmentation can be derived from the dense prediction of OCEs by using a post-processing step such as mean-shift clustering (fourth and eighth columns).

sis. Furthermore, to obtain more accurate cell segmentations, the segmentations generated in this unsupervised way can be used to augment very small amounts of manual labels and thus increase their efficacy without any additional costs (see Section 4.2). This will in turn drastically reduce the amount of human effort required to analyze large microscopy datasets, and provide a rich source of data for more quantitative and reproducible analyses.

However, we also note some limitations of our method stemming from violated assumptions: if objects are not randomly distributed (*e.g.*, if cells always cluster together in pairs), there is no way to tell in a purely unsupervised manner which structure is to be considered as one instance (either the pair of cells, or individual cells). Similarly, if the objects in the image do not resemble many other examples, the proposed method is unlikely to learn a meaningful object-centric embedding. As such, cells with outlier morphologies could result in degenerate segmentations. Furthermore, we note that the proposed method is sensitive to the size of the objects to be segmented, *i.e.*, the patch size has to be large enough to contain enough information to predict the relative position of the patch compared to others, but small enough to not contain entire objects. Although this introduces a hyper-parameter that has to be adjusted for each dataset, we believe that this is of little practical relevance since the size of cells in an image can easily be estimated.

While the work discussed here focuses on segmenting cells in 2D datasets, it is theoretically feasible to expand this method to 3D and even 4D datasets. This capability would be particularly useful to biological research, where cells and tissues are commonly imaged in 3D.

References

[1] Assaf Arbelle, Shaked Cohen, and Tammy Riklin Raviv. Dual-Task ConvLSTM-UNet for Instance Segmentation of Weakly Annotated Microscopy Videos. *IEEE Transactions on Medical Imaging*, 2022. 2

[2] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002. 5

[3] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic Instance Segmentation with a Discriminative Loss Function. *arXiv preprint arXiv:1708.02551*, 2017. 2

[4] Nizam Ud Din and Ji Yu. Training a deep learning model for single-cell segmentation without manual annotation. *Scientific reports*, 11(1):1–10, 2021. 3

[5] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2

[6] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[7] Noah F Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*, pages 1–11, 2021. 5, 7

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised Learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 2

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[10] Olivier Henaff. Data-Efficient Image Recognition with Contrastive Predictive Coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 3

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 11

[12] Manan Lalit, Pavel Tomancak, and Florian Jug. Embedding-based Instance Segmentation in Microscopy. In Mattias Heinrich, Qi Dou, Marleen de Bruijne, Jan Lellmann, Alexander Schläfer, and Floris Ernst, editors, *Proceedings of the Fourth Conference on Medical Imaging with Deep Learning*, volume 143 of *Proceedings of Machine Learning Research*, pages 399–415. PMLR, 07–09 Jul 2021. 1, 2

[13] Yiwen Li, Gratianus Wesley Putra Data, Yunguan Fu, Yipeng Hu, and Victor Adrian Prisacariu. Few-shot Semantic Segmentation with Self-supervision from Pseudo-classes. *arXiv preprint arXiv:2110.11742*, 2021. 3

[14] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. COMPLETER: Incomplete Multi-view Clustering via Contrastive Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11174–11183, 2021. 2

[15] Katharina Löffler and Ralf Mikut. EmbedTrack–Simultaneous Cell Segmentation and Tracking Through Learning Offsets and Clustering Bandwidths. *arXiv preprint arXiv:2204.10713*, 2022. 2

[16] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8837–8845, 2019. 2

[17] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[18] Christian Payer, Darko Štern, Marlies Feiner, Horst Bischof, and Martin Urschler. Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks. *Medical Image Analysis*, 57:106–119, 2019. 2

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 5, 7

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 11

[21] Tim Scherr, Katharina Löffler, Moritz Böhland, and Ralf Mikut. Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. *PLOS One*, 15(12):e0243219, 2020. 2

[22] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018. 2, 6, 7

[23] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021. 1, 2, 6

[24] Aiham Taleb, Winfried Loetzsch, Noel Danz, Julius Severin, Thomas Gaertner, Benjamin Bergner, and Christoph Lippert. 3D Self-Supervised Methods for Medical Imaging. *Advances in Neural Information Processing Systems*, 33:18158–18172, 2020. 3

[25] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised Learning from a Multi-view Perspective. *arXiv preprint arXiv:2006.05576*, 2020. 2

[26] Vladimír Ulman, Martin Maška, Klas EG Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, et al. An objective comparison of cell-tracking algorithms. *Nature Methods*, 14(12):1141–1152, 2017. 2, 5, 13, 14

[27] Steffen Wolf, Fred A. Hamprecht, and Jan Funke. Instance Separation Emerges from Inpainting, 2020. 3

[28] Xinpeng Xie, Jiawei Chen, Yuexiang Li, Linlin Shen, Kai Ma, and Yefeng Zheng. Instance-aware Self-supervised Learning for Nuclei Segmentation, 2020. 3

Appendix A. Self-Supervised Training

Architecture. Our self-supervised training requires a network (here referred to as mini U-Net) with a field of view (FoV) smaller than the expected cell diameter. Since our analyzed datasets contain cells with diameters as small as 20 pixels wide, we use a U-Net architecture [20] with an FoV of only 16×16. To increase the model's capabilities without expanding the FoV, we include additional 1×1 convolutions. Each U-Net block is composed of a series of $[3 \times 3, 1 \times 1, 1 \times 1, 3 \times 3]$ valid convolution layers with ReLU activations. We use a downsampling factor of 2×2, a depth of 1 and constant upsampling layers. In the first layer, we use 64 feature maps and increase it by a factor of 3 after each block.

Training. We train the mini U-Net on batches of 8 randomly chosen images with size 252×252 pixels. We use the Adam optimizer [11] with an initial learning rate of $4e^{-5}$ and train for 50 epochs, reducing the learning rate by a factor of 10 after epochs 20 and 30. In our pairwise loss, defined in Equation 5,

$$\mathcal{L} = \sum_{i,j \in P} \sigma\left(d(i,j) - \hat{d}(i,j)\right) + \lambda_{\text{reg}} \|r(i)\|_2,$$

we use $\sigma(\delta) = \left(1 + \exp\left(-\frac{\|\delta\|_2^2}{\tau}\right)\right)^{-1}$, $\tau = 10$, $\lambda_{\text{reg}} = 1e^{-5}$ and reduce the amount of coordinate pairs to P to reduce the GPU memory footprint. We obtain P by first sampling $\mathcal{P}_1$ as 10% of all pixels. For every sample in $p_1 \in \mathcal{P}_1$ we then sample $p_2 \in \mathcal{P}_2$, a random coordinate within radius $\kappa = 10$ of p_1. In our loss we sample $i, j \in P = \mathcal{P}_1 \times \mathcal{P}_2$.

Appendix B. Supervised Training

Architecture. We use the same architecture as the mini U-Net (see Appendix A) but increase the depth of the network to 3 which expands the network's field of view (FoV).

In conclusion, each U-Net block is composed of a series of $[3 \times 3, 1 \times 1, 1 \times 1, 3 \times 3]$ valid convolution layers with ReLU activations. We use a downsampling factor of 2×2, a depth of 3 and constant upsampling layers. In the first layer, we use 64 feature maps and increase it by a factor of 3 after each block.

Training. All models are trained with identical training setups. We optimize the loss (see $\mathcal{L}_{\text{STARDIST}}$) with the Adam optimizer with learning rate $1e^{-5}$ for 200 epochs, reducing the learning rate by a factor of 10 at epochs 30, 80 and 160. We use batches of 8 images with size 252×252 pixels, sampling pairs of patches within radius $\kappa = 10$ (see Equation 5), and set the loss temperature $\tau = 10$.

Appendix C. HSC Dataset

The HSC dataset is especially challenging - a culture plate is visible in the background, which causes all evaluated models to predict additional object instances near the border of the plate.

We compute scores on a range of IOU thresholds to investigate robustness of evaluated methods. At the matching IOU threshold of 0.5, we obtain F1 scores of 0.00 for the pre-trained CELLPOSE model, 0.09 for the pre-trained STARDIST model and 0.06 for CELLULUS *(ours)* (see Table 3). Additionally for the same matching IOU threshold, we obtain RECALL scores of 0.01 for CELLPOSE, 0.26 for STARDIST and 0.55 for CELLULUS. On the HSC dataset, CELLULUS is the most sensitive at detecting cells, but performs less favorably with respect to F1 and ACCURACY metrics. The high RECALL scores obtained with CELLULUS also explains the high SEG scores where false-positive predictions are not heavily penalized. Qualitative results on the HSC dataset can be seen in Figure 7.

Table 3. **Quantitative results on the HSC dataset in the Cell Tracking Challenge [26], for selected matching IOU thresholds (*fully unsupervised setting*).** Pre-trained CELLPOSE and STARDIST baseline models are compared with CELLULUS *(ours)*. Best performing method on each threshold and metric, is shown in bold.

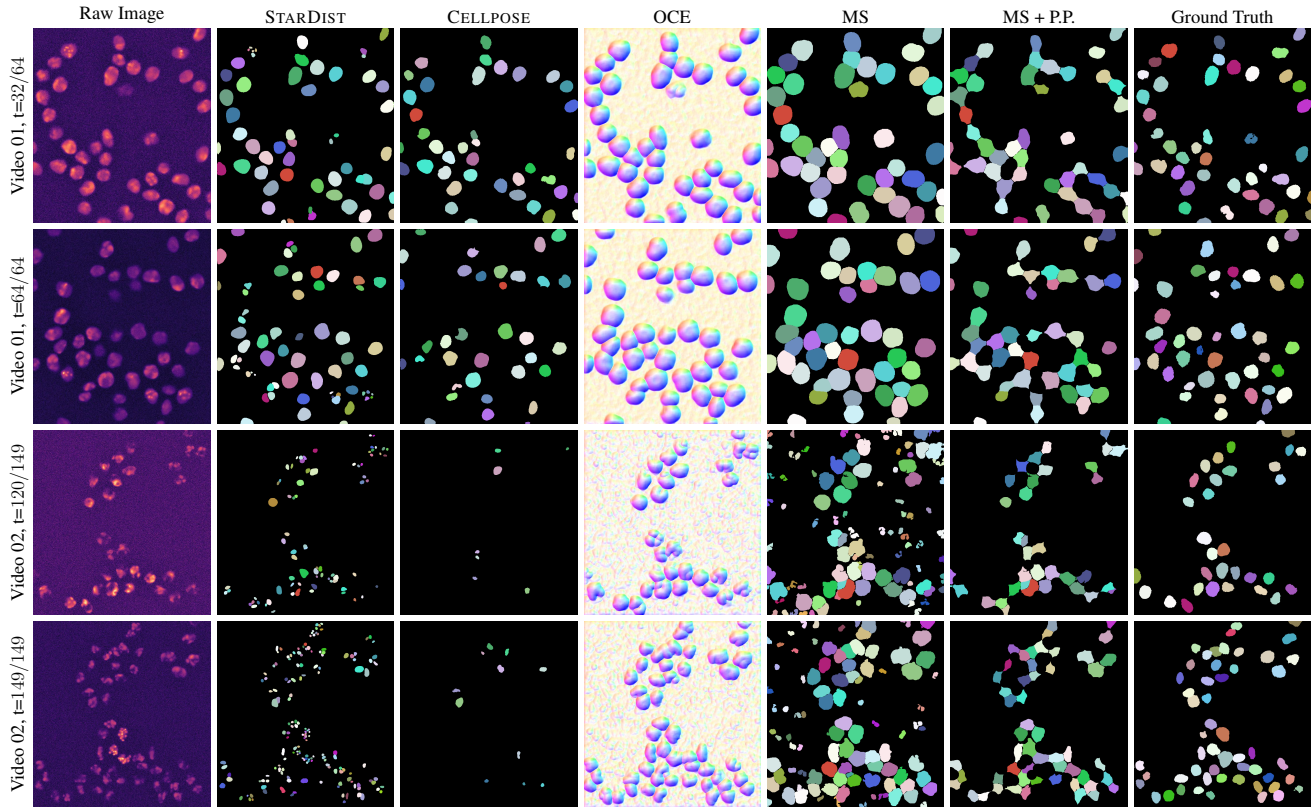| Threshold	ACCURACY			F1			RECALL		
	CELLPOSE	STARDIST	CELLULUS	CELLPOSE	STARDIST	CELLULUS	CELLPOSE	STARDIST	CELLULUS
0.1	**0.68**	0.22	0.06	**0.81**	0.36	0.11	0.92	0.98	**0.99**
0.2	**0.59**	0.21	0.06	**0.74**	0.35	0.11	0.84	**0.97**	0.96
0.3	**0.27**	0.18	0.05	**0.43**	0.30	0.10	0.51	0.83	**0.86**
0.4	0.03	**0.11**	0.04	0.06	**0.20**	0.08	0.07	0.55	**0.72**
0.5	0.00	**0.05**	0.03	0.00	**0.09**	0.06	0.01	0.26	**0.55**
0.6	0.00	0.01	**0.02**	0.00	0.03	**0.05**	0.00	0.07	**0.39**
0.7	0.00	0.00	**0.01**	0.00	0.00	**0.02**	0.00	0.01	**0.13**
0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.01**
0.9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Appendix D. SIMULATED Dataset

Figure 8. **Qualitative results on the SIMULATED dataset in the Cell Tracking Challenge [26]**. The SIMULATED dataset comprises of two time-lapse videos (Videos 01 and 02) which contain 64 and 149 image frames respectively. Shown here are individual raw images from the two videos (first column), predicted instance segmentations obtained using the pre-trained baseline models STARDIST (second column) and CELLPOSE (third column), dense prediction of Object-Centric Embeddings (OCEs) obtained using CELLULUS (fourth column), intermediate instance segmentations obtained by applying mean-shift (MS) clustering on the dense OCEs (fifth column), these intermediate instance segmentations are further post-processed (sixth column, see more details in the Section 4.1 - *Scale Informs All Parameter Choices*) and the Ground Truth Instance Segmentation available for evaluation purposes (seventh column).

Video 02 in SIMULATED (see last two rows) contains cells with visible granules which cause over-segmentation for both the evaluated, pre-trained baseline models.