Reviewed Preprint
v1 • September 6, 2024
Not revised

Cell Biology
Neuroscience

CellSeg3D: self-supervised 3D cell segmentation for microscopy

Cyril Achard, Timokleia Kousi, Markus Frey, Maxime Vidal, Yves Paychère, Colin Hofmann, Asim Iqbal, Sebastien B Hausmann, Stéphane Pagès, Mackenzie Weygandt Mathis ✉

Brain Mind Institute & Neuro X, École Polytechnique Fédérale de Lausanne (EPFL). Geneva, Switzerland • Wyss Center for Bio and Neuroengineering. Geneva, Switzerland

Abstract

Understanding the complex three-dimensional structure of cells is crucial across many disciplines in biology and especially in neuroscience. Here, we introduce a novel 3D self-supervised learning method designed to address the inherent complexity of quantifying cells in 3D volumes, often in cleared neural tissue. We offer a new 3D mesoSPIM dataset and show that CellSeg3D can match state-of-the-art supervised methods. Our contributions are made accessible through a Python package with full GUI integration in napari.

eLife assessment

This work presents a **valuable** new approach for self-supervised segmentation for fluorescence microscopy data, which could eliminate time-consuming data labeling and speed up quantitative analysis. The experimental evidence supplied is currently **incomplete** as the comparison with other methods is only done on a single dataset, and the usability of the Napari plugin is in question given the requirement of manual hyperparameter optimization.

https://doi.org/10.7554/eLife.99848.1.sa2

Main

Recent advancements in three-dimensional (3D) imaging techniques have provided unprecedented insights into cellular and tissue-level processes. In addition to confocal imaging and other fluorescent techniques, imaging systems based on light-sheet microscopy (LSM), such as the mesoscopic selective plane-illumination microscopy (mesoSPIM) initiative (1), have emerged as powerful tools for non-invasive, high-resolution 3D imaging of biological specimens. Due to its minimal phototoxicity and ability to capture high-resolution 3D images of thick biological samples, it has been a powerful new approach for imaging thick samples, such as the whole mouse brain, without the need for sectioning.

The analysis of such large-scale 3D datasets presents a significant challenge due to the size, complexity and heterogeneity of the samples. Yet, accurate and efficient segmentation of cells is a crucial step towards density estimates as well as quantification of morphological features. To begin to address this challenge, several studies have explored the use of supervised deep learning techniques using convolutional neural networks (CNNs) or transformers for improving cell segmentation accuracy (2–5). Various methods now exist for performing instance segmentation on the models" outputs in order to separate segmentation masks into individual cells.

Typically, these methods use a multi-step approach, first segmenting cells in 2D images, optionally performing instance segmentation, and then reconstructing them in 3D using the volume information (2, 3). While this can be successful in many contexts, this approach can suffer from low recall or have trouble retaining finer, non-convex labeling. Nevertheless, by training on (ideally large) human-annotated datasets, these supervised learning methods can learn to accurately segment cells in 2D, and ample 2D datasets now exist thanks to community efforts (6).

However, directly segmenting in 3D ("direct-3D") volumes could limit errors and streamline processing by retaining important morphological information. Yet, 3D annotated data is lacking (6), likely due to the fact that it is highly time-consuming to generate. For example, to our knowledge, 3D segmentation datasets of cells in whole-brain LSM volumes are not available, despite the existence of open-source microscopy database repositories (7).

Moreover, unsupervised learning, such as self-supervised learning, has emerged as a powerful approach for training deep neural networks without the need for explicit labeling of data. In the context of segmentation of cells, several studies have explored the use of unsupervised techniques to learn representations of cellular structures and improve segmentation accuracy (8, 9). However, these methods rely on adversarial learning, which can be difficult to train and have not been shown to provide accurate 3D results on cleared tissue for LSM data, which can suffer from clearing and artefacts.

Here, we present a new 3D dataset (**Figure 1a**) and custom toolbox for direct-3D supervised and self-supervised cell segmentation built on state-of-the-art transformers and 3D CNN architectures (10, 11) paired with classical image processing techniques (12). First, we benchmark our methods against Cellpose and StarDist, two leading supervised cell segmentation packages with user-friendly workflows, and show our methods match or outperform them in 3D instance segmentation on mesoSPIM-acquired volumes (**Figure 1b, c**). Then, we show that our self-supervised model, WNet3D, without any human labeled data can be as good as, or better than, supervised models (**Figure 1d-h**).

First, we developed a 3D human-annotated dataset based on data acquired with a mesoSPIM system (1) (**Figure 1a**, see Methods). Using whole-brain data from mice we cropped small regions and human annotated in 3D 2,632 neurons that were endogenously labeled by TPH2-tdTomato (**Figure 1a**).

We then trained two models for supervised direct-3D segmentation. Specifically, we used a SwinUNetR transformer (11), and a SegResNet CNN (13) from the MONAI project (14). We benchmarked these models against Cell-pose (3, 15) and StarDist (2) and find that our supervised models have comparable instance segmentation performance on held-out (unseen) test data set as measured by the F1 vs. IoU threshold; see Methods, **Figure 1b, c**). Note, for a fair comparison, we performed a hyperparameter sweep of all models tested (**Supplemental Figure S1a-d**), and in **Figure 1b** and **c** we show the quantitative and qualitative best models.

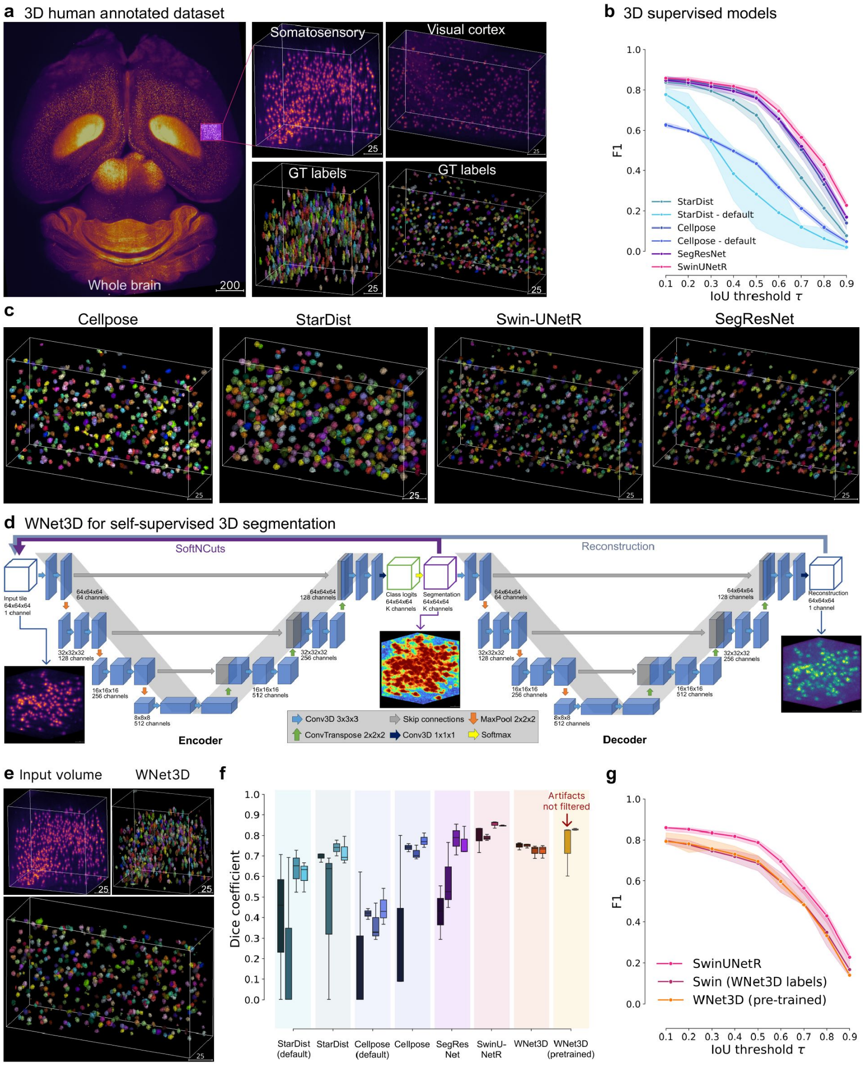


Figure 1.

Performance of 3D Semantic and Instance Segmentation Models.

a: Raw mesoSPIM whole-brain sample, volumes and corresponding ground truth labels from somatosensory (S1) and visual (V1) cortical regions. **b:** Evaluation of instance segmentation performance for several supervised models over three data subsets. F1-score is computed from the Intersection over Union (IoU) with ground truth labels, then averaged. Error bars represent 50% Confidence Intervals (CIs). **c:** View of 3D instance labels from supervised models, as noted, for visual cortex volume in **b** evaluation. **d:** Illustration of our WNet3D architecture showcasing the dual 3D U-Net structure with modifications (see Methods). **e:** Example 3D instance labels from WNet3D; top row is S1, bottom is V1, with artifacts removed. **f:** Semantic segmentation performance: comparison of model efficiency, indicating the volume of training data required to achieve a given performance level. Each supervised model was trained with an increasing percentage of training data (with 10, 20, 60 or 80%, left to right within each model grouping); Dice score was computed on unseen test data, over three data subsets for each training/evaluation split. Our self-supervised model (WNet3D) is also trained on a subset of the training set of images, but always without human labels. Far right: We also show performance of the pretrained WNet3D available in the plugin (far right), with and without removing artifacts in the image. See Methods for details. The central box represents the interquartile range (IQR) of values with the median as a horizontal line, the upper and lower limits the upper and lower quartiles. Whiskers extend to data points within 1.5 IQR of the quartiles. **g:** Instance segmentation performance comparison of Swin-UNetR and WNet3D (pretrained, see Methods), evaluated on unseen data across 3 data subsets, compared with a Swin-UNetR model trained using labels from the WNet3D self-supervised model. Here, WNet3D was trained on separate data, producing semantic labels that were then used to train a supervised Swin-UNetR model, still on held-out data. This supervised model was evaluated as the other models, on 3 held-out images from our dataset, unseen during training. Error bars indicate 50% CIs.

Next, we built a new self-supervised model for direct-3D segmentation that requires no ground truth training data, only raw volumes. Our new model, called WNet3D, is built on WNet (10) (see Methods, **Figure 1d**). Our changes include a conversion to a fully 3D architecture, adding the SoftNCuts loss, replacing the proposed two-step model update with the weighted sum of the encoder and decoder losses, and trimming the number of weights for fast inference (see Methods).

We found that WNet3D could be as good or better than the fully supervised models, especially in the low data regime, on this dataset at semantic and instance segmentation (**Figure 1e, f**). Notably, our pre-trained WNet3D, which is trained on 100% of raw data without any labels, achieves 0.81±0.004 Dice coefficient with simple filtering of artifacts (removing the slices containing the problematic regions) and 0.74±0.12 without any filtering. To compare, we trained supervised models with 10, 20, 60 or 80% of the training data and tested on the held-out data subsets. Considering models with 80% of the training data, the Dice coefficient for SwinUNetR was 0.83±0.01, 0.76±0.03 for Cellpose tuned, 0.74±0.06 for SegResNet, 0.72±0.07 for StarDist (tuned), 0.61±0.07 for StarDist (default), 0.43±0.09 for Cellpose (default). For WNet3D with 80% raw data for training was 0.71±0.03 (un-filtered) (**Figure 1f**), which is still on-par with the top supervised models.

Notably, for models with only 10% of the training data, the Dice coefficient was 0.78 ± 0.07 for SwinUNetR, 0.69 ± 0.02 for StarDist (tuned), 0.42 ± 0.13 for SegResNet, 0.39 ± 0.36 for StarDist (default), 0.32 ± 0.4 for Cellpose tuned, 0.20 ± 0.35 for Cellpose (default), and WNet3D was 0.74 ± 0.02 (unfiltered), which is still on-par with the top supervised model, and much improved (2X) over most supervised base-lines, most strikingly at low-data regimes (**Figure 1f**).

Thus, over the four data subsets we tested (**Supplemental Figure S1e**), we find significant differences in model performance (Kruskal-Wallis H test, H=49.21, p=2.06e-08, n=12). With post-hoc Conover-Iman testing, WNet3D showed significant performance gains over StarDist and Cellpose (defaults) (statistics in **Supplemental Figure S1f**). More importantly, it is not significantly different from the best performing models (i.e., SwinUNetR p=1, and other competitive supervised models: Cellpose (tuned) p=0.21, or Seg-ResNet p=0.076; **Supplemental Figure S1f**). Altogether, our self-supervised model can perform as well as top supervised approaches.

Note that WNet3D uses brightness to detect objects, and therefore cannot discriminate cells vs artifacts. Filtering could be used when sufficient (e.g., using rules based on label volume to remove aberrantly small or large particles), or one could use WNet3D to generate 3D labels in order to train a suitable supervised model (such as Cellpose or SwinUNetR), which would be able to distinguish artifacts from cells.

To show the feasibility of this approach, we trained a Swin-UNetR using WNet3D self-supervised generated labels (**Figure 1g**) and show it can be nearly as good as a fully supervised model that required human 3D labels (no significant difference across F1 vs. IoU thresholds; Kruskal-Wallis H test H=4.91, p=0.085, n=9).

Lastly, we highlight that the models we present are available in a new napari plugin we developed, with full support for labeling, training (self-supervised or supervised), model inference, and evaluation plus many other utilities (**Figure 2a**). Moreover, our pretrained WNet3D can be used “zero-shot” on diverse data, such as Platynereis nuclei, mouse skull bone nuclei (both collected with confocal microscopy; (**Figure 2b-c**), even though qualitatively these datasets are quite distinct looking from the dataset used for pretraining. We also tested the pretrained WNet3D on c-FOS stained tissue, which had more difficult signal to noise due to clearing and anti-body staining, from whole brains of mice acquired with a mesoSPIM (**Figure 2d**).

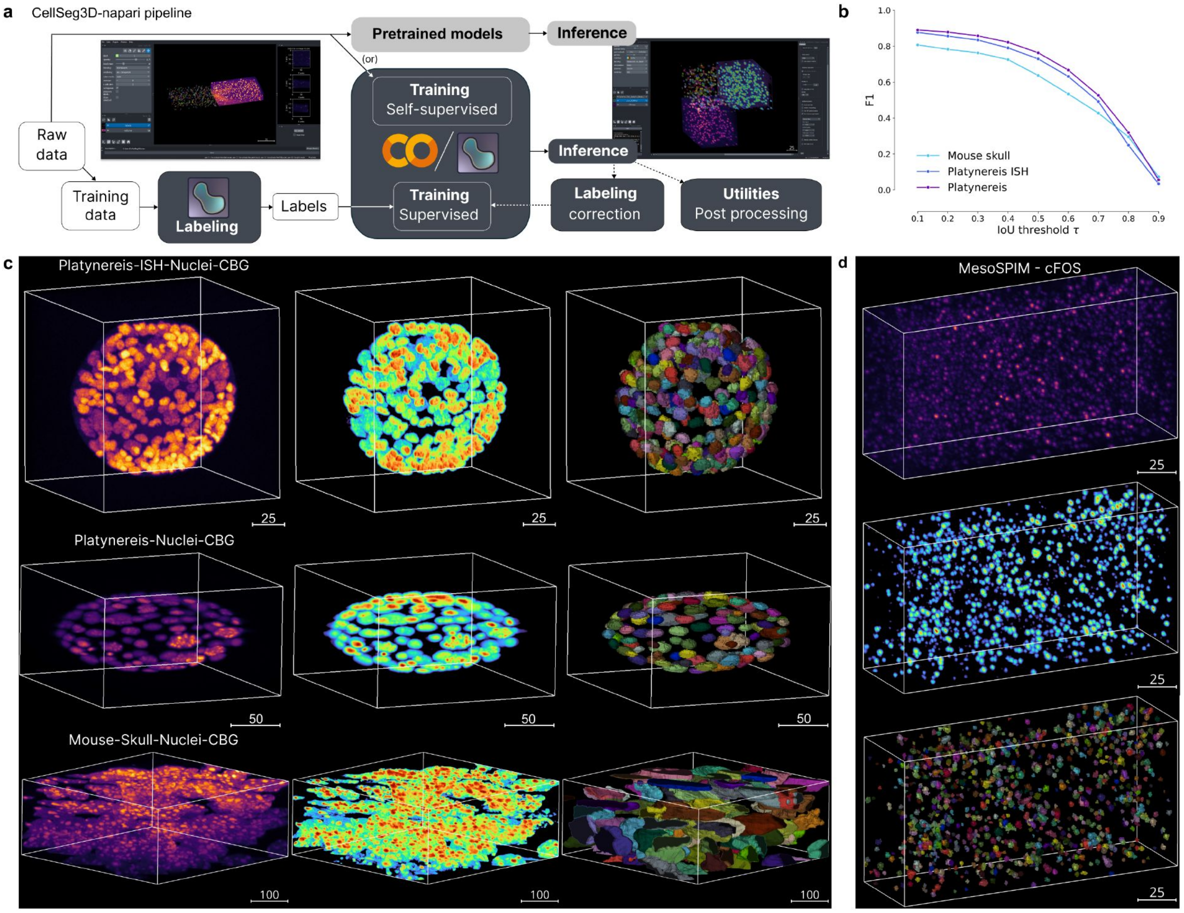


Figure 2.

CellSeg3D napari plugin pipeline, training, and example outputs.

a: Workflow diagram depicting the segmentation pipeline: either raw data can be used directly (self-supervised) or labeled and used for training and then other data can be used for model inference. Each stream concludes with posthoc inspection and refinement, if needed (post-processing analysis and/or refining the model). **b:** Instance segmentation performance (zero-shot) of the pretrained WNet3D on select datasets featured in **c**, shown as F1-score vs IoU with ground truth labels. **c:** Qualitative examples with WNet3D for semantic and instance segmentation. **d:** Qualitative example of WNet3D-generated prediction (thresholded) and labels on a crop from a whole-brain sample, with c-FOS-labeled neurons, acquired with a mesoSPIM.

In summary, CellSeg3D supports high-performance supervised and self-supervised direct-3D segmentation. Our napari plugin supports both the pretrained WNet3D and ability to train it and other models presented here (SegRes-Net, SwinUNetR), and has various tools for pre- and post-processing as well as utilities for labeling with minimal effort. We additionally provide our new 3D dataset intended for benchmarking 3D cell segmentation algorithms on LSM acquired cleared-tissue (see Data Card), and all code is fully open-sourced at *https://github.com/AdaptiveMotorControlLab/CellSeg3D*.

Acknowledgements

The authors thank Martin Weigert, Jessy Lauer and members of the Mathis Lab for inputs and comments on the manuscript.

M.W.M. acknowledges the Wyss Institute for partly funding this work, and the Bertarelli Foundation.

Author Contributions Statement

Conceptualization: C.A., M.W.M.; Methodology: C.A., M.V., M.F.; Software: C.A., M.V., C.H., Y.P., T.K.; Investigation: C.A.; Dataset Acquisition: S.B.H, T.K., S.P.; Dataset Labeling: T.K.; Writing-Original Draft: C.A., M.W.M., T.K., M.F.; Supervision: M.W.M., M.F., A.I.; Funding Acquisition: M.W.M.

Declaration of interests

The authors declare no competing interests.

Dataset Availability

Labeled 3D data is available at: *https://zenodo.org/records/11095111*;see our Supplemental Data Card.

Code Availability

All code is available at: *https://github.com/AdaptiveMotorControlLab/CellSeg3D* and code to reproduce the Figures is available at: *https://github.com/C-Achard/CellSeg3D-figures*.

Methods

Datasets

CellSeg3D LSM dataset: acquisition and labeling

The whole-brain data by Voigt et al. (1) was obtained from the IDR platform (7); the volume consists of CLARITY cleared tissue from a TPH2-tdTomato mouse. Data was acquired with the mesoSPIM system at a zoom of 0.63X with 561 nm excitation.

The data was cropped to several regions of the somatosensory (5 volumes, without artifacts) and visual cortex (1 volume, with artifacts) and annotated by an expert. The ground-truth cell count for the dataset is as follows:

Additional datasets

Additional datasets, used in **Figure 2c** were taken from the *GitHub page* of EmbedSeg, by (16). We used our pretrained WNet3D, without re-training (the model was only trained on our new dataset described above), to generate semantic segmentation. The channel containing the foreground was then thresholded and the Voronoi-Otsu algorithm used to generate instance labels (for Platynereis data), with hyperparameters based on the Dice metric with the ground truth. However, these parameters can also be estimated directly.

For the Mouse Skull Nuclei instance segmentation, we performed additional post-processing using clEsperanto (12) to perform a morphological closing operation with radius 8 on semantic labels in order to remove small holes. The image was then remapped to values ∈[0; 100] for convenience, before merging labels with a touching border within intensity range between 35 and 100 using the *merge_labels_with_border_intensity_within_range* function.

For **Figure 2d**, we used a wild type C57BL/6J adult mouse (17 weeks old, Female) that was given appetitive food 90 min before deep anesthesia and intra-cardial perfusion with 4% PFA. We followed establish guidelines for iDISCO (17). In brief, the brain was dehydrated, bleached, permeabilized and stained for c-FOS using anti-c-FOS Rat monoclonal purified IgG (Synaptic Systems, Cat. No. 226 017) followed by a Donkey anti-Rat IgG Alexa Fluor− 555 (Invitrogen A78945) secondary antibody.

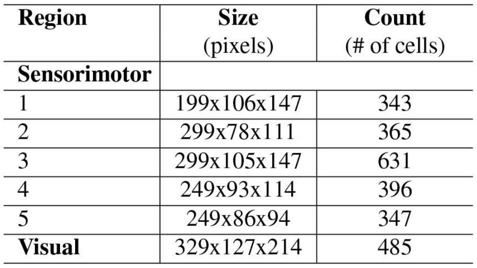
Then, the whole brain was imaged on a mesoSPIM (1). Imaging was performed with a laser at a wavelength of 561 nm, with a pixel size of 5.26×5.26 μm in x,y, and a step size of 5 μm in z. All experimental protocols adhered to the stringent ethical standards set forth by the Veterinary Department of the Canton Geneva, Switzerland, with all procedures receiving approval and conducted under license number 33020 (GE10A).

Segmentation models and algorithms: Self-supervised semantic segmentation

WNet3D model architecture

To perform self-supervised cell segmentation, we adapted the WNet architecture proposed by Xia and Kulis (10), an autoencoder architecture based on joining two U-Net models end-to-end. We provide a modified version of the WNet, named WNet3D, with the following changes:

- A conversion of the architecture for fully-3D segmentation, including the SoftNCuts loss
- Replacing the proposed two-step model update with the weighted sum of the encoder and decoder losses, updated in a single backward pass
- Reducing the overall depth of the encoder and decoder, using three up/downsampling steps instead of four



Region	**Size** (pixels)	**Count** (# of cells)
Sensorimotor		
1	199x106x147	343
2	299x78x111	365
3	299x105x147	631
4	249x93x114	396
5	249x86x94	347
Visual	329x127x214	485

Table 1.

Dataset ground-truth cell count per volume.

- Replacing batch normalization with group normalization, tuning the number of groups based on performance

Reducing the number of layers improved overall performance by reducing overfitting and sped up training and inference. This trimming was meant to reduce the large number of parameters resulting from a naive conversion of the original WNet architecture to 3D, which were found to be unnecessary for the present cell segmentation task. Finally, we introduced group normalization([18]) to replace batch normalization, which improved performance in the present low batch size setting, as well as training and inference speed.

To summarize, the model consists of an encoder U_{enc} and decoder U_{dec}, as originally proposed; however, each UNet comprises 7 blocks, for a total of 14 blocks, down from 9 blocks per UNet originally. U_{enc} and U_{dec} start and end with 2 3 × 3 × 3 3D convolutional layers, in-between are 5 blocks, each block being defined by two 3×3×3 3D convolutional layers, followed by a ReLU and group normalization ([18]) (instead of batch normalization). Skip connections are used to propagate information by concatenating the output of descending blocks to that of their corresponding ascending blocks. Blocks are followed by 2×2×2 max pooling layers in the descending half of U_{enc} and U_{dec}, the ascending half uses 2×2×2 transpose convolution layers with stride= 2 ; U_{enc} is then followed by a 1× 1× 13D convolutional layer to obtain class logits, follwed by a softmax, the output of which is provided to U_{dec} to perform the reconstruction. U_{dec} is similarly followed by a 1×1×13D convolutional layer and outputs the reconstructed volume. Refer to figure (**Figure 1d**) for a complete overview of the WNet3D architecture.

Losses

Segmentation is performed in U_{enc} by using the an adapted 3D SoftNCuts loss as an objective, with the voxel brightness differences defining the edge weight in the calculation, as proposed in the initial Ncuts algorithm by Shi and Malik ([19]).

The SoftNCuts is defined as

$$Ncut_K(V) = \sum_{k=1}^{K} \frac{cut(A_k, V - A_k)}{cut(A_k, V)} \qquad \textbf{(1)}$$

where $cut(A, B) = \Sigma_{u \in A, v \in B} w(u, v)$, V is the set of all pixels, A_k the set of all pixels labeled as class k and $w(u, v)$ is the weight of the edge uv in a graph representation of the image; in order to group the voxels according to brightness, $w(u, v)$ is defined here as

$$w(u,v) = \\ e^{\frac{-\|\mathbf{F}(u)-\mathbf{F}(v)\|_2^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|\mathbf{X}(u)-\mathbf{X}(v)\|_2^2}{\sigma_X}} & \text{if } \|\mathbf{X}(u)-\mathbf{X}(v)\| < r \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{(2)}$$

with $F(i) = I(i)$ the intensity value, σ_I the standard deviation of the feature similarity term, termed “intensity sigma”, σ_X the standard deviation of the spatial proximity term, termed “spatial sigma”, and r the radius for the calculation of the loss, to avoid computing every pairwise value.

In our experiments, lowering the radius greatly sped up training without impacting performance, even with a radius as low as 2 voxels. For the spatial sigma, the original value of 4 was used, whereas for the intensity sigma we use a value of 1 (originally 4), after remapping voxel values in each image to the [0; 100] range.

U_{dec} then uses a suitable reconstruction loss to reconstruct the original image; we used either Mean Squared Error (MSE) or Binary Cross Entropy (BCE) as defined in PyTorch.

Hyperparameters

To achieve proper cell segmentation, it was crucial to prevent the SoftNCuts from simply separating the data in broad regions with differing overall brightness; this was achieved by adjusting the weighting of the reconstruction loss accordingly. In our experiments, we empirically adapted the weights to equalize the contribution of each loss term, making sure we have uniform gradients in the backward pass. This proved effective for training on our provided dataset; however, for different samples, adjusting the reconstruction weight and learning rate using the ranges specified below was necessary for good performance; other parameters were kept constant.

The default number of classes is two, to segment background and cells, but this number may be raised to add more brightness-grouped classes; this could be useful to mitigate the over-segmentation of cells due to brightness “halos” surrounding the nucleus, or to help produce labels for object boundary segmentation.

We found that summing the losses, instead of iteratively updating the encoder first followed by the whole network as suggested, improved stability and consistency of loss convergence during training; in our version the trade-off between accuracy of reconstruction and quality of segmentation is controlled by adjusting the parameters of the weighted sum instead of individual learning rates.

This modified model was usually trained for 50 epochs, unless stated otherwise. We use a batch size of 2, 2 classes, a radius of 2 for the NCuts loss and the MSE reconstruction loss, and use a learning rate between $2 \cdot 10^{-3}$ and $2 \cdot 10^{-5}$ and reconstruction loss weight between $5 \cdot 10^{-3}$ and $5 \cdot 10^{-1}$, depending on the data.

See **Supplemental Figure S2a** for an overview of the training process, including loss curves and model outputs.

Segmentation models and algorithms: Supervised semantic segmentation

Model architectures

In order to perform supervised fully-3D cell segmentation, we leveraged computer vision models and losses implemented by the MONAI project, which offers several state-of-the-art architectures. The MONAI API was used as the basis for our napari plugin, and we retained two of the provided models based on their performance on the provided dataset:

- SegResNet (13)
- SwinUNetR (11)

SegResNet is based on the Convolutional Neural Network (CNN) architecture, whereas SwinUNetR uses a transformer-based encoder.

Several relevant segmentation losses are made available for training:

- Dice loss (20)
- Dice-Cross Entropy loss
- Generalized Dice loss (21)
- Tversky loss (22)

The SegResNet and SwinUNetR models shown here were trained using the Generalized Dice loss for 50 epochs, with a learning rate of $1{\cdot}10^{-3}$, batch size of 5 (SwinUNetR) or 10 (SegResNet), and data augmentation enabled. Unless stated otherwise, a train/test split of 80/20% was used.

The outputs were then passed through a threshold to discard low-confidence predictions; this was estimated using the training set to find the threshold that maximized the Dice metric between predictions and ground truth. The same process was repeated for Cellpose (cell probability threshold) and StarDist (non-maximum suppression (NMS) and cell probability thresholds) to ensure fair comparisons, see "Model comparison" below and **Supplemental Figure S1a,b,c,d** for tuning results. Inference outputs are processed a-posteriori to obtain instance labels, as detailed below.

Instance segmentation

Several methods for instance segmentation are available in the plugin: the connected components and watershed algorithms (scikit-image), and the Voronoi-Otsu labeling method (clEsperanto). The latter combines an Otsu threshold and a Voronoi tessellation to perform instance segmentation, and more readily avoids fusing clumped cells than the former two, provided that the objects are spherical, which is the case in the present task.

The Voronoi-Otsu method was therefore used to perform instance segmentation in the benchmarks, with its two parameters, spatial sigma and outline sigma, tuned to fit the training data when relevant, and manually selected otherwise.

Model Comparisons

StarDist was retrained using the provided example notebook for 3D, using default parameters. For the model we refer to as "Default", we used a patch size of 8x64x64, a grid of (2,1,1), a batch size of 2 and 96 rays, as computed automatically in the provided example code for StarDist. For the "Tuned" version (referred to simply as "StarDist"), we changed the patch size to 64x64x64 and the grid to (1,1,1).

Cellpose was retrained without pretrained weights using default parameters, except for the mean diameter which was set to 3.3 according to the provided object size estimation utility. We investigated all pretrained models provided by Cellpose, as well as attempting transfer learning, but no pretrained model was found to be suitable for our data. "Default" refers to automatically estimated parameters for StarDist (NMS and probability threshold, estimated on the training data), and cell probability threshold of 0 with resampling enabled for Cellpose. For both models, inference hyperparameters (respectively NMS and cell probability threshold for StarDist and cell probability threshold and resampling on CellPose) were tuned on the training set to maximize the Dice metric with GT labels, exactly like our models. After tuning, we found that Cellpose achieved best performance with a cell probability threshold of −9 and resampling enabled (see **Supplemental Figure S1a**) across all data subset. For StarDist, best parameters varied across subsets (see **Supplemental Figure S1d**), however, as this did not affect performance significantly, we used the parameters estimated automatically as part of the training.

Models provided in the plugin (SwinUNetR, SegResNet and WNet3D), which we refer to as "pretrained", are trained on the entire dataset, using all images (and labels only for the supervised models). The WNet3D model was used in **Figure 1f** (WNet3D - pretrained), g (WNet3D pretrained and SwinUnetR), and **Figure2b** (WNet3D). Hyperparameters used are as mentioned above, except for the number of epochs, which was selected based on validation curves.

Label efficiency comparison

To assess how many labeled cells are required to reach a certain performance, we trained StarDist, Cellpose, SegResNet, SwinUNetR and WNet on three distinct subsets of the data, each time holding out one full volume of the full dataset for evaluation, fragmenting the remaining volumes and labels into 64 pixels cubes, and training on distinct train/validation splits on remaining data. We used 10%, 20%, 60% and 80% splits in order to assess how much labeled data is necessary for the supervised models, and whether they show variability based on the data used for training. To note, the evaluation data remained the same for all percentages in a given data subset, ensuring a consistent performance comparison. We used 50 epochs for all runs, and no early stopping or hyperparameter tuning was performed based on the validation performance during training. Instead, we reused the best hyperparameters found for **Figure 1b**.

For example, the first subset consists of all five somatosensory cortex volumes as training/validation data, and the visual cortex volume is held out for evaluation. For Cellpose two conditions are shown, default (cell probability threshold of 0) and fine-tuned (threshold of -9), which improved performance.

To avoid training on data with artifacts present in the visual cortex volume, WNet3D was only trained on the first of the subsets. Instead, the model was trained on a percentage of the first subset using three different seeds. We also avoid evaluating on artifacts in the visual volume, unless mentioned otherwise, as the model is not meant to handle these regions.

WNet3D-based retraining of supervised models

To assess whether WNet3D can generalize to unseen data when trained on a specific brain volume, we trained a WNet3D from scratch using volumes cropped from a different mesoSPIM-acquired whole brain sample, labeled with c-FOS, imaged at 561 nm with a pixel size of 5.26 × 5.26μm in x and y, and a step size in z of 5μm (see Additional Datasets).

This model was then used to generate labels for our provided dataset. A SwinUNetR model was then trained using these WNet3D generated labels, and compared to the performance of the pretrained model we provide in our napari plugin.

Performance evaluation

Instance segmentation

Model performance was evaluated and compared via the matching dataset utilities provided by StarDist (2). Briefly, several accuracy metrics are computed as functions of several overlap thresholds τ ; true positives are pairings of model predictions and ground-truth labels having an intersection over union (IoU) value greater than the specified threshold, with automated matching to prevent additional instances from being assigned to the same ground truth or model-predicted instance of a label. The thresholds for the 3D models were chosen based on the Dice metric between training labels and model-generated labels, unless specified otherwise.

Semantic segmentation

The Dice-Sørensen coefficient was used to evaluate semantic segmentation performance, defined as

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}. \tag{3}$$

CellSeg3D napari plugin workflow

To facilitate the use of our models, we provide a napari plugin where users can easily annotate data, train models, run inference, and perform various post-processing steps. Starting from raw data, users can quickly crop regions into regions of interest, and create training data from those. Users may manually annotate the data in napari using our labeling interface, which provides additional interface such as orthogonal projections to better view the ongoing labeling process, as well as keeping track of time spent labeling each slice, or alternatively train a self-supervised model to automatically perform a first iteration of the segmentation and labeling, without annotation. Users can also try pretrained models, including the self-supervised one, to generate labels which can then be corrected using the same labeling interface. Supervised or self-supervised models can then be trained using the generated data. Full documentation for the plugin can be found on our GitHub website.

In the case of supervised learning, the volumes (random patches or whole images) are split into training and validation sets according to a user-set proportion, using 80%/20% by default. Input images are normalized by setting all values above and below the 1st and 99th percentile to the corresponding percentile value, respectively. Data augmentation can be used; by default a random shift of the intensity, elastic and affine deformations, flipping and rotation are used.

For the self-supervised model, images are remapped to values in the [0;100] range to accommodate the intensity sigma of the SoftNCuts loss. No percentile normalization is used and data augmentation is restricted to flipping and rotating in this case.

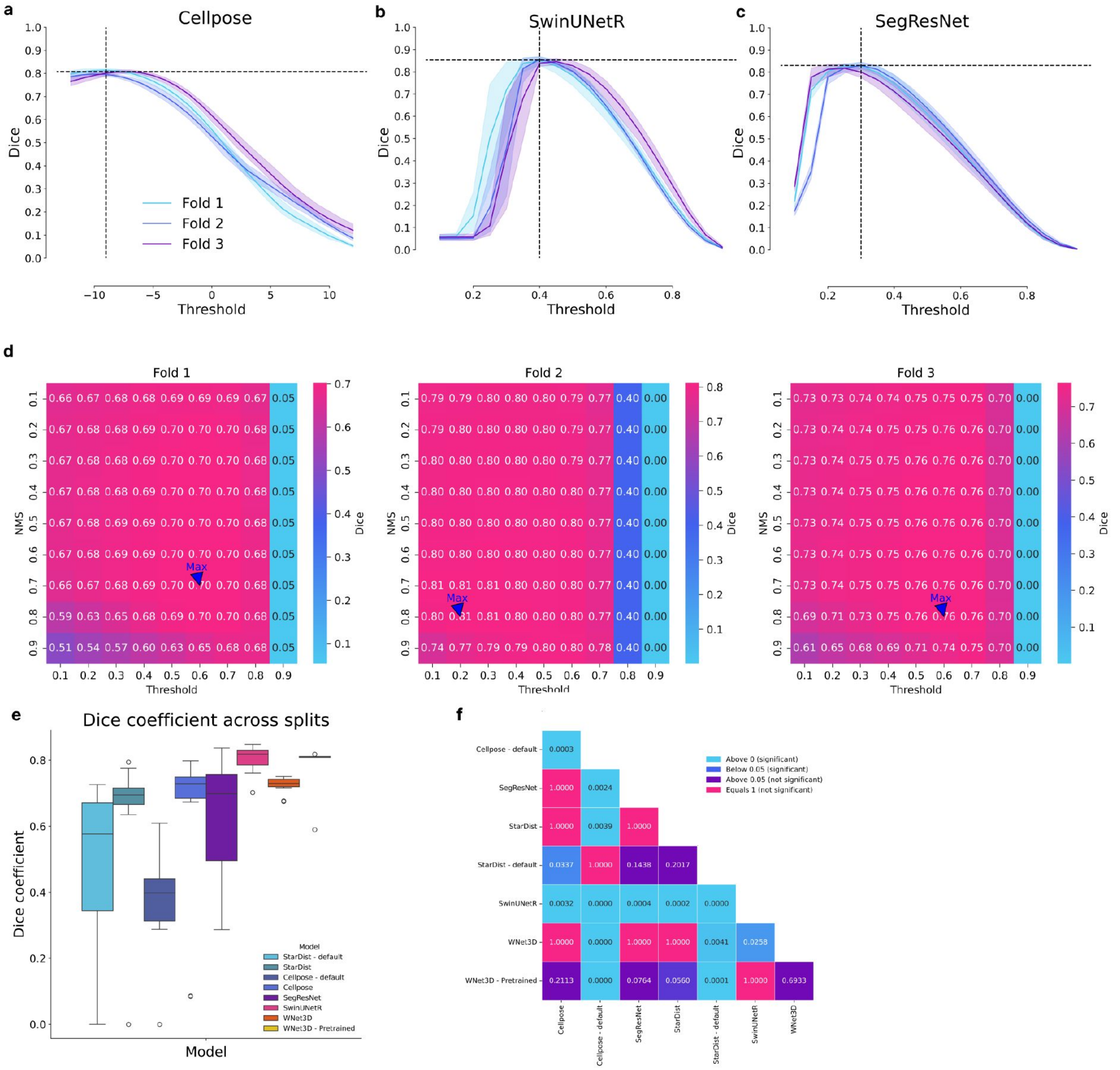
Deterministic training may also be enabled for all models and the random generation seed set; unless specified otherwise, models were trained on cropped cubes with 64 pixels edges, with both data augmentation and deterministic training enabled.

We additionally provide a Colab notebook to train our self-supervised model using the same procedure described above. The pretrained weights for all our models are also made available through the HuggingFace platform (and automatically downloaded by the plugin or in Colab), so that users without the recommended hardware can still easily train or try our models. All code is open source and available on GitHub.

Statistical Methods

To confirm whether there were statistically significant differences in model performance, we pooled accuracy values (across IoU for **Figure 1b**, and **g** and **Figure 2b**, and across percentage of training data used for **Figure 1f**) and in Python 3.8 using the scikit_posthocs package we performed a Kruskal-Wallis test to check the null hypothesis that the median of all models was equal. When this test was significant, we used two-sided Conover-Iman post-hoc testing to test pairwise differences between models, also using the "scikit_posthoc" implementation, with the Holm-Bonferroni correction for multiple comparisons (step-down method using Bonferroni adjustments).

Supplemental Information



Fold 1

NMS \ Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.66	0.67	0.68	0.68	0.69	0.69	0.69	0.67	0.05
0.2	0.67	0.68	0.68	0.69	0.70	0.70	0.70	0.68	0.05
0.3	0.67	0.68	0.68	0.69	0.70	0.70	0.70	0.68	0.05
0.4	0.67	0.68	0.68	0.69	0.70	0.70	0.70	0.68	0.05
0.5	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.68	0.05
0.6	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.68	0.05
0.7	0.66	0.67	0.68	0.69	0.70	0.70	0.70	0.68	0.05
0.8	0.59	0.63	0.65	0.68	0.69	0.70	0.70	0.68	0.05
0.9	0.51	0.54	0.57	0.60	0.63	0.65	0.68	0.68	0.05

Fold 2

NMS \ Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.79	0.79	0.80	0.80	0.80	0.79	0.77	0.40	0.00
0.2	0.79	0.80	0.80	0.80	0.80	0.79	0.77	0.40	0.00
0.3	0.80	0.80	0.80	0.80	0.80	0.79	0.77	0.40	0.00
0.4	0.80	0.80	0.80	0.80	0.80	0.80	0.77	0.40	0.00
0.5	0.80	0.80	0.80	0.80	0.80	0.80	0.77	0.40	0.00
0.6	0.80	0.80	0.80	0.80	0.80	0.80	0.77	0.40	0.00
0.7	0.81	0.81	0.81	0.80	0.80	0.80	0.77	0.40	0.00
0.8	0.80	0.81	0.81	0.80	0.80	0.80	0.77	0.40	0.00
0.9	0.74	0.77	0.79	0.79	0.80	0.80	0.78	0.40	0.00

Fold 3

NMS \ Threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.73	0.73	0.74	0.74	0.75	0.75	0.75	0.70	0.00
0.2	0.73	0.74	0.74	0.75	0.75	0.76	0.76	0.70	0.00
0.3	0.73	0.74	0.75	0.75	0.76	0.76	0.76	0.70	0.00
0.4	0.73	0.74	0.75	0.75	0.76	0.76	0.76	0.70	0.00
0.5	0.73	0.74	0.75	0.75	0.76	0.76	0.76	0.70	0.00
0.6	0.73	0.74	0.75	0.75	0.76	0.76	0.76	0.70	0.00
0.7	0.73	0.74	0.75	0.75	0.76	0.76	0.76	0.70	0.00
0.8	0.69	0.71	0.73	0.75	0.76	0.76	0.76	0.70	0.00
0.9	0.61	0.65	0.68	0.69	0.71	0.74	0.75	0.70	0.00

	Cellpose	Cellpose - default	SegResNet	StarDist	StarDist - default	SwinUNetR	WNet3D
Cellpose - default	0.0003						
SegResNet	1.0000	0.0024					
StarDist	1.0000	0.0039	1.0000				
StarDist - default	0.0337	1.0000	0.1438	0.2017			
SwinUNetR	0.0032	0.0000	0.0004	0.0002	0.0000		
WNet3D	1.0000	0.0000	1.0000	1.0000	0.0041	0.0258	
WNet3D - Pretrained	0.2113	0.0000	0.0764	0.0560	0.0001	1.0000	0.6933

Figure S1.

Hyperparameter tuning of baselines and statistics

a,b,c: Hyperparameter optimisation for several supervised models. In Cellpose, the cell probability threshold value is applied before the sigmoid, hence values between −12 and 12 were tested. CellSeg3D models return predictions between 0 and 1 after applying the softmax, values tested were therefore in this range. Error bars show 95% CIs. **d:** StarDist hyperparameter optimisation. Several parameters were tested for non-maximum suppression (NMS) threshold and cell probability threshold. **e:** Pooled Dice scores per split, related to **Figure 1f**, used for statistical testing shown in **f**. The central box represents the interquartile range (IQR) of values with the median as a horizontal line, the upper and lower limits the upper and lower quartiles. Whiskers extend to data points within 1.5 IQR of the quartiles. Outliers are shown separately. **f:** Pairwise Conover's test p-values for the Dice metric values per model shown in **e**. Colors are based on level of significance.

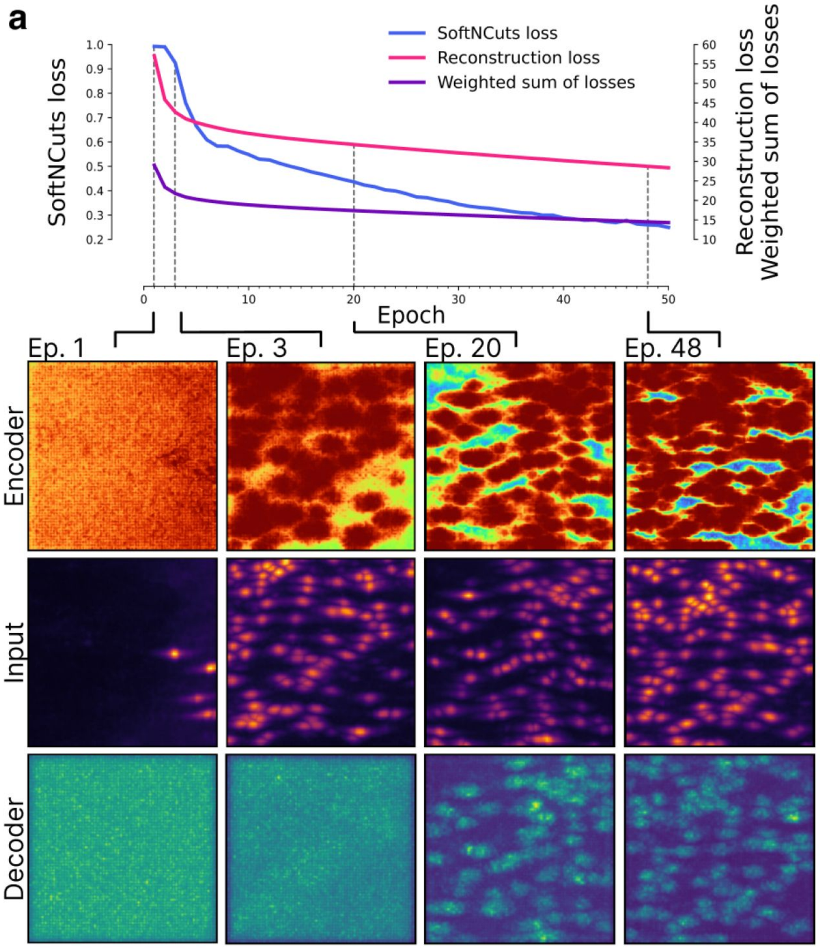


Figure S2.

Training WNet3D

a: Overview of the training process of WNet3D. The loss for the encoder U_{enc} is the SoftNCuts, whereas the reconstruction loss for U_{dec} is MSE. The weighted sum of losses is calculated as indicated in Methods. For select epochs, input volumes are shown, with outputs from encoder U_{enc} above, and outputs from decoder U_{dec} below.

Dataset Card

A. Motivation

1. *For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description*. The contributions of our dataset to the vision and cell biology communities are twofold: 1) We release a 3D cell segmentation dataset of 2632 TPH2 positive cells, based on data from Voigt et al.(1). 2) The dataset is one of the first cell segmentation datasets to date created in 3D. It aims to advance cell segmentation research in neuroscience and vision communities.
2. *Who created the dataset (which team, research group) and on behalf of which entity (company, institution, organization)?* The annotated dataset was created by the Mathis Lab of Adaptive Intelligence of EPFL. The raw brain data is publicly available on *https://idr.openmicroscopy.org/webclient/?show=project-854*.
3. *Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number*. This project was funded, in part, by the Wyss Center via a grant to PI Mathis.
4. *Any other comments?* No.

Composition

1. *What do the instances that comprise the dataset represent (e.g.,documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description*. The instances in our dataset represent 3D volumetric segments, extracted from mesoSPIM scans of mouse brains. Each instance is essentially a three-dimensional image that has been carefully cropped mainly from the somatosensory and visual cortex of the scanned data. In each of these 3D volumes, TPH2 cells are identified and labeled.
2. *How many instances are there in total (of each type, if appropriate)?* There are six 3D volumetric segments, that contain a total of 2638 TPH2 positive cells identified and labeled in 3D.
3. *Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? If so, please describe how this representativeness was validated/verified. If it is not representative of the larger set, please describe why not (e.g., to cover a more diverse range of instances, because instances were withheld or unavailable)*. The dataset provided is a subset of the available whole-brain sample, selected from larger raw volumetric data obtained from mesoSPIM scans of mouse brains. This selection primarily consists of 3D volumes cropped mainly from the somatosensory and visual cortex regions, where the TPH2 cells are labeled meticulously. The broader dataset from which these instances were extracted represents scans of whole mouse brains. However, due to the immense volume of the entire scanned data, creating a manageable and focused dataset was key for addressing specific research questions and computational manageability.
4. *What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description*. Each instance in the dataset consists of "raw" 3D volumetric data derived from mesoSPIM scans of mouse brains, specifically focusing on the somatosensory cortex and vision cortex regions. The instances are essentially unprocessed and maintain the integrity of the original scanned data.

5. *Is there a label or target associated with each instance? If so, please provide a description*. Yes, each instance in the dataset is annotated with masks. There are no categories or text associated with the masks.
6. *Is any information missing from individual instances? If so,please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text*. In our dataset, there is no information missing from individual instances.
7. *Are relationships between individual instances made explicit (e.g., users" movie ratings, social network links)? If so, please describe how these relationships are made explicit*. Not applicable.
8. *Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description*. While we have taken extensive measures to ensure the accuracy and quality of the dataset, it is challenging to rule out the presence of minor errors or noise, especially considering the complex nature of the 3D cell segmentation task. Nonetheless, we believe that any such inconsistencies do not compromise the overall reliability and utility of the dataset.
9. *Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, a) are there guarantees that they will exist, and remain constant, over time; b) are there official archival versions of the complete dataset (i.e., including the external resources as they existed at the time the dataset was created); c) are there any restrictions (e.g., licenses, fees) associated with any of the external resources that might apply to a dataset consumer? Please provide descriptions of all external resources and any restrictions associated with them, as well as links or other access points, as appropriate*. The dataset is self-contained.
10. *Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals" non-public communications)? If so, please provide a description*. No.
11. *Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why*. No. The dataset is composed solely on scientific, non-human biological data.
12. *Does the dataset identify any subpopulations (e.g., by age, gender)? If so, please describe how these subpopulations are identified and provide a description of their respective distributions within the dataset*. Not applicable.
13. *Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how*. Not applicable.
14. *Does the dataset contain data that might be considered sensitive in anyway (e.g., data that reveals race or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description*. No.
15. *Any other comments?* No.

Collection Process

1. *How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If the data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how*. The data associated with each instance was acquired through mesoSPIM scans of mouse brains, providing raw, directly observable 3D volumetric data. The data was not reported by subjects or indirectly inferred or derived from other data; it was directly observed and recorded from the scientific imaging process. All collected volumes were annotated by expert human annotators. The quality of the annotations was validated by an external expert not involved in the annotation process.
2. *What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? How were these mechanisms or procedures validated?* The raw data is open source and provided by the Image Data Resource (IDR).
3. *If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?* Our sampling strategy was designed to select volumes where TPH2 cells are clearly discernible. We aimed to include a varied range of volumes, from densely packed with TPH2 cells to ones more sparsely populated, ensuring a good representation of various brain areas. Another important factor was the manageability of the volumes from an annotation perspective, to facilitate accurate and efficient labeling.
4. *Who was involved in the data collection process(e.g.,students,crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?* The released masks were created by research personnel of the Mathis Lab of Adaptive Intelligence, EPFL.
5. *Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created*. The raw data was downloaded from the Image Data Resource (IDR) website. The labels were created between June and October 2021.

If the dataset does not relate to people, you may skip the remaining questions in this section

Preprocessing / Cleaning / Labeling

1. *Was any preprocessing / cleaning / labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remaining questions in this section*. Yes, extensive preprocessing, and labeling were conducted to ensure the usability and reliability of the dataset. The initial step involved examination of the raw 3D volumetric data, where we ruled out the presence of anomalies or artefacts. During this phase, we ensured the visibility of TPH2-positive cells within the volumetric segments. We proceeded to label the TPH2-positive cells through a well-defined annotation process, where each cell within the selected volumes was identified and marked by our experts. At the end of the annotation process, the quality of the work was verified by a human expert not involved in the annotation work.

2. *Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data*. The raw data is open source and available on the Image Data Resource (IDR) website.
3. *Is the software that was used to preprocess/clean/label the data available? If so, please provide a link or other access point*. Yes. We used the napari interactive viewer for multidimensional images in Python.

Uses

1. *Has the dataset been used for any tasks already? If so, please provide a description*. The dataset was used to train our segmentation models.
2. *Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point*. Yes, the repository hosting the model weights which were trained on our data, as well as the repository for our napari plugin for 3D cell segmentation.
3. *What (other) tasks could the dataset be used for?* We intend the dataset to be used to train cell segmentation models. However, we invite the research community to gather additional annotations for mesoSPIM acquired datasets via the tools we contribute in the present publication.
4. *Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a dataset consumer might need to know to avoid uses that could result in unfair treatment of individuals or groups (e.g., stereotyping, quality of service issues) or other risks or harms (e.g., legal risks, financial harms)? If so, please provide a description. Is there anything a dataset consumer could do to mitigate these risks or harms?* Not applicable.
5. *Are there tasks for which the dataset should not be used? If so, please provide a description*. Full terms of use for the dataset can be found at *https://github.com/AdaptiveMotorControlLab/CellSeg3D*, but the project is made open source under an MIT license.

Distribution

1. *Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description*. The dataset is released on zenodo at: *https://zenodo.org/records/11095111*.
2. *How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)?* The dataset is released on zenodo at: *https://zenodo.org/records/11095111*.
3. *When will the dataset be distributed?* The dataset is released on zenodo at: *https://zenodo.org/records/11095111* alongside the publication of this paper.
4. *Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions*. The dataset is released under a MIT license.
5. *Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions*. Full terms of use and restrictions on use of the provided 3D cell segmentation dataset can be found at *https://github.com/AdaptiveMotorControlLab/CellSeg3D*.

6. *Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation.* The dataset is released under a MIT license.
7. *Any other comments?* No.

Maintenance

1. *Who will be supporting/hosting/maintaining the dataset?* The dataset will be hosted at *https://github.com/AdaptiveMotorControlLab/CellSeg3D* and maintained by the Mathis Lab of Adaptive Intelligence.
2. *How can the owner/curator/manager of the dataset be contacted(e.g.,email address)?* Please see contact information at *https://github.com/AdaptiveMotorControlLab/CellSeg3D*.
3. *Is there an erratum? If so, please provide a link or other access point*. No.
4. *Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to dataset consumers (e.g., mailing list, GitHub)?* To ensure reproducibility of research this dataset won"t be updated. Any issues or errors will be publicly shared.
5. *If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced*. Not applicable.
6. *Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to dataset consumers*. This is the first version.
7. *If others want to extend/augment/build on/contribute to the dataset,is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Is there a process for communicating/distributing these contributions to dataset consumers? If so, please provide a description*. We warmly encourage users to enhance the value of this project by contributing additional annotations and annotated datasets. If you have relevant data, please consider sharing them by linking the data to our GitHub repository. For any inquiries, suggestions, or discussions related to the project, please feel free to reach out to us on GitHub *https://github.com/AdaptiveMotorControlLab/CellSeg3D*.
8. *Any other comments?* No.

B. Data Annotation Card

Task Formulation

1. *At a high level, what are the subjective aspects of your task?* Object segmentation within an image is a subjective task (23). Distinguishing between structures that represent cells and artifacts relies on the annotator"s judgment and expertise. This can lead to variability in the quality and quantity of the masks generated per image by different annotators. To mitigate this risk we engaged experts from our research lab, to annotate the volumes. We insisted on the quality of annotations over their quantity; we aimed to annotate smaller volumes to ensure accurate representation of the cell nuclei, even if it meant having fewer annotations.

2. *What assumptions do you make about annotators?* Our annotator is a member of our research lab, ensuring a close understanding of the project"s goals. The team concentrated on two main objectives. 1) Clear Understanding of Project Goals: We worked to fully understand the project"s aims and translated them into clear and straightforward guidelines, which included visual examples. 2) Regular Sharing of Updates and Results: we reviewed our aims and results to make ongoing improvements to the annotation process. This regular check-in helped in quickly addressing any issues and adding new material to improve our annotation quality.
3. *How did you choose the specific wording of your task instructions? What steps, if any, were taken to verify the clarity of task instructions and wording for annotators?* The annotator was a co-creator of the annotation instructions and guidelines, which boosted their understanding. As our task was annotations images, we crafted visual examples with step by step instructions. We collectively decide how to handle complex and unambiguous cases, and refine the guidelines throughout the process. The project team met for feedback and updates, while the annotator was able to give feedback on an asynchronous way at any time.
4. *What, if any,risks did your task pose for annotators and were they informed of the risks prior to engagement with the task?* No identified risks.
5. *What are the precise instructions that were provided to annotators?* We created clear guides on installing and using the napari annotation tool. The task was to segment every TPH2 positive cell in a given image. The annotator created a 3D mask for each cell they identified, using the tool to precisely add or remove areas of the mask around the cell. In simpler terms, they had to isolate each cell in 3D using the tool, making sure it was accurate down to the pixel-level.

Selecting Annotations

1. *Are there certain perspectives that should be privileged? If so, how did you seek these perspectives out?* We chose to engage researchers that have a deep understanding on cell biology and vision research.
2. *Are there certain perspectives that would be harmful to include? If so, how did you screen these perspectives out?* No.
3. *Were sociodemographic characteristics used to select annotators for your task? If so, please detail the process*. No.
4. *If you have any aggregated socio-demographic statistics about your annotator pool, please describe. Do you have reason to believe that sociode-mographic characteristics of annotators may have impacted how they annotated the data? Why or why not?* Our annotator worked in our research institute.
5. *Consider the intended context of use of the dataset and the individuals and communities that may be impacted by a model trained on this dataset. Are these communities represented in your annotator pool?* Not applicable.

Platform and Infrastructure Choices

- *What annotation platform did you utilize? At a high level, what considerations informed your decision to choose this platform? Did the chosen platform sufficiently meet the requirements you outlined for annotator pools? Are any aspects not covered?* We used napari, a fast, interactive viewer for multi-dimensional images in Python. Link: *https://napari.org/stable/*
- *What, if any, communication channels did your chosen platform offer to facilitate communication with annotators? How did this channel of communication influence the annotation process and/or resulting annotations?* Communication was established through other internal communication platforms.

- *How much were annotators compensated? Did you consider any particular pay standards, when determining their compensation?* If so, please describe. The compensation was based on their employment contract at EPFL.

Dataset Analysis and Evaluation

1. *How do you define the quality of annotations in your context, and how did you assess the quality in the dataset you constructed?* To assess the quality of the annotations in the constructed dataset, we included a review process. Annotations were created by an expert well-acquainted with the morphological characteristics of TPH2 positive cells, ensuring a high level of initial accuracy. Any ambiguous cases in annotation were resolved through discussions amongst the team until a consensus was reached. Regular feedback was provided to the annotator, and any identified errors or inconsistencies were promptly corrected.
2. *Have you conducted any analysis on disagreement patterns? If so, what analyses did you use and what were the major findings? Did you analyze potential sources of disagreement?* We provided regular feedback sessions in a synchronous and asynchronous way.
3. *How do the individual annotator responses relate to the final labels released in the dataset?* Our dataset along with our annotations are available and accessible through zenodo: *https://zenodo.org/records/11095111*.

Dataset Release and Maintenance

1. *Do you have reason to believe the annotations in this dataset may change over time? Do you plan to update your dataset?* No.
2. *Are there any conditions or definitions that, if changed, could impact the utility of your dataset?* We do not believe so.
3. *Will you attempt to track, impose limitations on, or otherwise influence how your dataset is used? If so, how?* No.
4. *Were annotators informed about how the data is externalized? If changes to the dataset are made, will they be informed?* Yes.
5. *Is there a process by which annotators can later choose to withdraw their data from the dataset? If so, please detail*. No.

References

1. Voigt Fabian F. *et al.* (2019) **The mesoSPIM initiative: open-source light-sheet microscopes for imaging cleared tissue** *Nature Methods* :1105–1108 https://doi.org/10.1038/s41592-019-0554-0

2. Weigert Martin, Schmidt Uwe, Haase Robert, Sugawara Ko, Myers Gene (2020) **Star-convex polyhedra for 3d object detection and segmentation in microscopy** *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* :3666–3673

3. Stringer Carsen, Wang Tim, Michaelos Michalis, Pachitariu Marius (2021) **Cellpose: a generalist algorithm for cellular segmentation** *Nature Methods* :100–106 https://doi.org/10.1038/s41592-020-01018-x

4. Iqbal Asim, Sheikh Asfandyar, Karayannis Theofanis (2019) **Denerd: high-throughput detection of neurons for brain-wide analysis with deep learning** *Scientific Reports* **9**

5. Hörst Fabian *et al.* (2023) **CellViT: Vision Transformers for Precise Cell Segmentation and Classification** *arXiv* https://doi.org/10.48550/arXiv.2306.15350

6. Ma Jun *et al.* (2024) **The multimodality cell segmentation challenge: toward universal solutions** *Nature Methods* :1–11 https://doi.org/10.1038/s41592-024-02233-6

7. Williams Eleanor *et al.* (2017) **Image Data Resource: a bioimage data integration and publication platform** *Nature Methods* :775–781 https://doi.org/10.1038/nmeth.4326

8. Yao Kai, Sun Jie, Huang Kaizhu, Jing Linzhi, Liu Hang, Huang Dejian, Jude Curran (2021) **Analyzing Cell-Scaffold Interaction through Unsupervised 3D Nuclei Segmentation** *International Journal of Bioprinting* **8** https://doi.org/10.18063/ijb.v8i1.495

9. Han Liang, Yin Zhaozheng, de Bruijne Marleen, Cattin Philippe C., Cotin Stéphane, Padoy Nicolas, Speidel Stefanie, Zheng Yefeng, Essert Caroline (2021) **Unsupervised Network Learning for Cell Segmentation** *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021, Lecture Notes in Computer Science* :282–292 https://doi.org/10.1007/978-3-030-87193-2_27

10. Xia Xide, Kulis Brian (2017) **W-Net: A Deep Model for Fully Unsupervised Image Segmentation** *arXiv* https://doi.org/10.48550/arXiv.1711.08506

11. Hatamizadeh Ali, Tang Yucheng, Nath Vishwesh, Yang Dong, Myronenko Andriy, Landman Bennett, Roth Holger, Xu Daguang (2021) **UNETR: Transformers for 3D Medical Image Segmentation** *arXiv* https://doi.org/10.48550/arXiv.2103.10504

12. Haase Robert *et al.* (2020) **CLIJ: GPU-accelerated image processing for everyone** *Nature Methods* :5–6 https://doi.org/10.1038/s41592-019-0650-1

13. Myronenko Andriy (2018) **3D MRI brain tumor segmentation using autoen-coder regularization** *arXiv* https://doi.org/10.48550/arXiv.1810.11654

14. The MONAI Consortium. Project monai (2020) **The MONAI Consortium. Project monai. Zenodo, December 2020. doi: 10.5281/zenodo.4323059.** https://doi.org/10.5281/zenodo.4323059

15. Stringer Carsen, Pachitariu Marius (2022) **Cellpose 2.0: how to train your own model** *Nature Methods* **19**:1634–1641

16. Lalit Manan, Tomancak Pavel, Jug Florian (2021) **Embedding-based instance segmentation of microscopy images** *arXiv* https://doi.org/10.48550/arXiv.2101.10033

17. Renier Nicolas, Wu Zhuhao, Simon David J, Yang Jing, Ariel Pablo, Tessier-Lavigne Marc (2014) **idisco: a simple, rapid method to immunolabel large tissue samples for volume imaging** *Cell* **159**:896–910

18. Wu Yuxin, He Kaiming (2018) **Group Normalization** *arXiv* https://doi.org/10.48550/arXiv.1803.08494

19. Shi Jianbo, Malik J. (2000) **Normalized cuts and image segmentation** *IEEE Transactions on Pattern Analysis and Machine Intelligence* :888–905 https://doi.org/10.1109/34.868688

20. Milletari Fausto, Navab Nassir, Ahmadi Seyed-Ahmad (2016) **V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation** *arXiv* https://doi.org/10.48550/arXiv.1606.04797

21. Sudre Carole H., Li Wenqi, Vercauteren Tom, Ourselin Sébastien, Cardoso M. Jorge (2017) **Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations** *arXiv* https://doi.org/10.48550/arXiv.1707.03237

22. Salehi Seyed Sadegh Mohseni, Erdogmus Deniz, Gholipour Ali (2017) **Tversky loss function for image segmentation using 3D fully convolutional deep networks** *arXiv* https://doi.org/10.48550/arXiv.1706.05721

23. Kirillov Alexander *et al.* (2023) **Segment anything** *arXiv* https://doi.org/10.48550/arXiv.2304.02643

Editors

Reviewing Editor
Albert Cardona
University of Cambridge, Cambridge, United Kingdom

Senior Editor
Albert Cardona
University of Cambridge, Cambridge, United Kingdom

Reviewer #1 (Public Review):

This work makes several contributions: (1) a method for the self-supervised segmentation of cells in 3D microscopy images, (2) an cell-segmented dataset comprising six volumes from a mesoSPIM sample of a mouse brain, and (3) a napari plugin to apply and train the proposed method.

(1) Method

This work presents itself as a generalizable method contribution with a wide scope: self-supervised 3D cell segmentation in microscopy images. My main critique is that there is almost no evidence for the proposed method to have that wide of a scope. Instead, the paper is more akin to a case report that shows that a particular self-supervised method is good enough to segment cells in two datasets with specific properties.

To support the claim that their method "address[es] the inherent complexity of quantifying cells in 3D volumes", the method should be evaluated in a comprehensive study including different kinds of light and electron microscopy images, different markers, and resolutions to cover the diversity of microscopy images that both title and abstract are alluding to.

The main dataset used here (a mesoSPIM dataset of a whole mouse brain) features well-isolated cells that are easily distinguishable from the background. Otsu thresholding followed by a connected component analysis already segments most of those cells correctly. The proposed method relies on an intensity-based segmentation method (a soft version of a normalized cut) and has at least five free parameters (radius, intensity, and spatial sigma for SoftNCut, as well as a morphological closing radius, and a merge threshold for touching cells in the post-processing). Given the benefit of tweaking parameters (like thresholds, morphological operation radii, and expected object sizes), it would be illuminating to know how other non-learning-based methods will compare on this dataset, especially if given the same treatment of segmentation post-processing that the proposed method receives. After inspecting the WNet3D predictions (using the napari plugin) on the used datasets I find them almost identical to the raw intensity values, casting doubt as to whether the high segmentation accuracy is really due to the self-supervised learning or instead a function of the post-processing pipeline after thresholding.

I suggest the following baselines be included to better understand how much of the segmentation accuracy is due to parameter tweaking on the considered datasets versus a novel method contribution:
* comparison to thresholding (with the same post-processing as the proposed method)
* comparison to a normalized cut segmentation (with the same post-processing as the proposed method)
* comparison to references 8 and 9.

I further strongly encourage the authors to discuss the limitations of their method. From what I understand, the proposed method works only on well-separated objects (due to the semantic segmentation bottleneck), is based on contrastive FG/BG intensity values (due to the SoftNCut loss), and requires tuning of a few parameters (which might be challenging if no ground-truth is available).

(2) Dataset

I commend the authors for providing ground-truth labels for more than 2500 cells. I would appreciate it if the Methods section could mention how exactly the cells were labelled. I found a good overlap between the ground truth and Otsu thresholding of the intensity images. Was the ground truth generated by proofreading an initial automatic segmentation, or entirely done by hand? If the former, which method was used to generate the initial segmentation, and are there any concerns that the ground truth might be biased towards a given segmentation method?

(3) Napari plugin

The plugin is well-documented and works by following the installation instructions. However, I was not able to recreate the segmentations reported in the paper with the default settings for the pre-trained WNet3D: segments are generally too large and there are a lot of false positives. Both the prediction and the final instance segmentation also show substantial border artifacts, possibly due to a block-wise processing scheme.

https://doi.org/10.7554/eLife.99848.1.sa1

Reviewer #2 (Public Review):

Summary:

The authors propose a new method for self-supervised learning of 3d semantic segmentation for fluorescence microscopy. It is based on a WNet architecture (Encoder / Decoder using a UNet for each of these components) that reconstructs the image data after binarization in the bottleneck with a soft n-cuts clustering. They annotate a new dataset for nucleus segmentation in mesoSPIM imaging and train their model on this dataset. They create a napari plugin that provides access to this model and provides additional functionality for training of own models (both supervised and self-supervised), data labeling, and instance segmentation via post-processing of the semantic model predictions. This plugin also provides access to models trained on the contributed dataset in a supervised fashion.

Strengths:

(1) The idea behind the self-supervised learning loss is interesting.

(2) The paper addresses an important challenge. Data annotation is very time-consuming for 3d microscopy data, so a self-supervised method that yields similar results to supervised segmentation would provide massive benefits.

Weaknesses:

The experiments presented by the authors do not adequately support the claims made in the paper. There are several shortcomings in the design of the experiment and presentation of the results. Further, it is unclear if results of similar quality as reported can be achieved within the GUI by non-expert users.

Major weaknesses:

(1) The main experiments are conducted on the new mesoSPIM dataset, which contains quite small and well separated nuclei. It is unclear if the good performance of the novel self-supervised learning method compared to CellPose and StarDist would hold for dataset with other characteristics, such as larger nuclei with a more complex morphology or crowded nuclei. Further, additional preprocessing of the mesoSPIM images may improve results for StarDist and CellPose (see the first point in minor weaknesses). Note: having a method that works better for small nuclei would be an important contribution. But I am uncertain the claims hold for larger and/or more crowded nuclei as the current version of the paper implies. The contribution of the paper would be stronger if a comparison with StarDist / CellPose was also done on the additional datasets from Figure 2.

(2) The experimental setup for the additional datasets seems to be unrealistic. In general, the description of these experiments is quite short and so the exact strategy is unclear from the text. However, you write the following: "The channel containing the foreground was then thresholded and the Voronoi-Otsu algorithm used to generate instance labels (for Platynereis data), with hyperparameters based on the Dice metric with the ground truth." I.e., the hyperparameters for the post-processing are found based on the ground truth. From the description it is unclear whether this is done a) on the part of the data that is then also used to compute metrics or b) on a separate validation split that is not used to compute metrics. If

a): this is not a valid experimental setup and amounts to training on your test set. If b): this is ok from an experimental point of view, but likely still significantly overestimates the quality of predictions that can be achieved by manual tuning of these hyperparameters by a user that is not themselves a developer of this plugin or an absolute expert in classical image analysis, see also 3. Note that the paper provides notebooks to reproduce the experimental results. This is very laudable, but I believe that a more extended description of the experiments in the text would still be very helpful to understand the set-up for the reader. Further, from inspection of these notebooks it becomes clear that hyper-parameters where indeed found on the testset (a), so the results are not valid in the current form.

(3) I cannot obtain similar results to the ones reported in the manuscript using the plugin. I tried to obtain some of the results from the paper qualitatively: First I downloaded one of the volumes from the mesoSPIM dataset (c5image) and applied the WNet3D to it. The prediction looks ok, however the value range is quite narrow (Average BG intensity ~0.4, FG intensity 0.6-0.7). I try to apply the instance segmentation using "Convert to instance labels" from "Utilities". Using "Voronoi-Otsu" does not work due to an error in pyClesperanto ("clGetPlatformIDs failed: PLATFORM_NOT_FOUND_KHR"). Segmentation via "Connected Components" and "Watershed" requires extensive manual tuning to get a somewhat decent result, which is still far from perfect.

Then I tried to obtain the results for the Mouse Skull Nuclei Dataset from EmbedSeg. The results look like a denoised version of the input image, not a semantic segmentation. I was skeptical from the beginning that the method would transfer without retraining, due to the very different morphology of nuclei (much larger and elongated). None of the available segmentation methods yield a good result, the best I can achieve is a strong over-segmentation with watersheds.

Minor weaknesses:

(1) CellPose can work better if images are resized so that the median object size in new images matches the training data. For CellPose the cyto2 model should do this automatically. It would be important to report if this was done, and if not would be advisable to check if this can improve results.

(2) It is a bit confusing that F1-Score and Dice Score are used interchangeably to evaluate results. The dice score only evaluates semantic predictions, whereas F1-Score evaluates the actual instance segmentation results. I would advise to only use F1-Score, which is the more appropriate metric. For Figure 1f either the mean F1 score over thresholds or F1 @ 0.5 could be reported. Furthermore, I would advise adopting the recommendations on metric reporting from https://www.nature.com/articles/s41592-023-01942-8.

(3) A more conceptual limitation is that the (self-supervised) method is limited to intensity-based segmentation, and so will not be able to work for cases where structures cannot be distinguished based on intensity only. It is further unclear how well it can separate crowded nuclei. While some object separation can be achieved by morphological operations this is generally limited for crowded segmentation tasks and the main motivation behind the segmentation objective used in StarDist, CellPose, and other instance segmentation methods. This limitation is only superficially acknowledged in "Note that WNet3D uses brightness to detect objects [...]" but should be discussed in more depth.

Note: this limitation does not mean at all that the underlying contribution is not significant, but I think it is important to address this in more detail so that potential users know where the method is applicable and where it isn't.

https://doi.org/10.7554/eLife.99848.1.sa0

Author Response:

> ***Reviewer #1 (Public Review):***
>
> *This work makes several contributions: (1) a method for the self-supervised segmentation of cells in 3D microscopy images, (2) an cell-segmented dataset comprising six volumes from a mesoSPIM sample of a mouse brain, and (3) a napari plugin to apply and train the proposed method.*

First, thanks for acknowledging our contributions of a new tool, new dataset, and new software.

> *(1) Method*
>
> *This work presents itself as a generalizable method contribution with a wide scope: self-supervised 3D cell segmentation in microscopy images. My main critique is that there is almost no evidence for the proposed method to have that wide of a scope. Instead, the paper is more akin to a case report that shows that a particular self-supervised method is good enough to segment cells in two datasets with specific properties.*

First, thanks for acknowledging our contributions of a new tool, new dataset, and new software. We agree we focus on lightsheet microscopy data, therefore to narrow the scope we have changed the title to “CellSeg3D: self-supervised 3D cell segmentation for **light-sheet** microscopy”.

> *To support the claim that their method "address[es] the inherent complexity of quantifying cells in 3D volumes", the method should be evaluated in a comprehensive study including different kinds of light and electron microscopy images, different markers, and resolutions to cover the diversity of microscopy images that both title and abstract are alluding to. The main dataset used here (a mesoSPIM dataset of a whole mouse brain) features well-isolated cells that are easily distinguishable from the background. Otsu thresholding followed by a connected component analysis already segments most of those cells correctly.*

You have selectively dropped the last part of that sentence that is key: “.... 3D volumes, **often in cleared neural tissue**” – which *is* what we tackle. The next sentence goes on to say: “We offer a new 3D mesoSPIM dataset and show that CellSeg3D can match state-of-the-art supervised methods.” Thus, we literally make it clear our claims are on MesoSPIM and cleared data.

> *The proposed method relies on an intensity-based segmentation method (a soft version of a normalized cut) and has at least five free parameters (radius, intensity, and spatial sigma for SoftNCut, as well as a morphological closing radius, and a merge threshold for touching cells in the post-processing). Given the benefit of tweaking parameters (like thresholds, morphological operation radii, and expected object sizes), it would be illuminating to know how other non-learning-based methods will compare on this dataset, especially if given the same treatment of segmentation post-processing that the proposed method receives. After inspecting the WNet3D predictions (using the napari plugin) on the used datasets I find them almost identical to the raw intensity values, casting doubt as to whether the high segmentation accuracy is really due to the self-supervised learning or instead a function of the post-processing pipeline after thresholding.*

First, thanks for testing our tool, and glad it works for you. The deep learning methods we use cannot "solve" this dataset, and we also have a F1-Score (dice) of ~0.8 with our self-supervised method. We don't see the value in applying non-learning methods; this is unnecessary and beyond the scope of this work.

> *I suggest the following baselines be included to better understand how much of the segmentation accuracy is due to parameter tweaking on the considered datasets versus a novel method contribution:*
> ** comparison to thresholding (with the same post-processing as the proposed method)*
> ** comparison to a normalized cut segmentation (with the same post-processing as the proposed method)*
> ** comparison to references 8 and 9.*

Ref 8 and 9 don't have readily usable (https://github.com/LiangHann/USAR) or even shared code (https://github.com/Kaiseem/AD-GAN), so re-implementing this work is well beyond the bounds of this paper. We benchmarked Cellpose, StartDist, SegResNets, and a transformer – SwinURNet. Moreover, models in the MONAI package can be used. Note, to our knowledge the transformer results also are a new contribution that the Reviewer does not acknowledge.

> *I further strongly encourage the authors to discuss the limitations of their method. From what I understand, the proposed method works only on well-separated objects (due to the semantic segmentation bottleneck), is based on contrastive FG/BG intensity values (due to the SoftNCut loss), and requires tuning of a few parameters (which might be challenging if no ground-truth is available).*

We added text on limitations. Thanks for this suggestion.

> *(2) Dataset*
>
> *I commend the authors for providing ground-truth labels for more than 2500 cells. I would appreciate it if the Methods section could mention how exactly the cells were labelled. I found a good overlap between the ground truth and Otsu thresholding of the intensity images. Was the ground truth generated by proofreading an initial automatic segmentation, or entirely done by hand? If the former, which method was used to generate the initial segmentation, and are there any concerns that the ground truth might be biased towards a given segmentation method?*

In the already submitted version, we have a 5-page DataSet card that fully answers your questions. They are ALL labeled by hand, without any semi-automatic process.

In our main text we even stated "Using whole-brain data from mice we cropped small regions and human annotated in 3D 2,632 neurons that were endogenously labeled by TPH2-tdTomato" - clearly mentioning it is human-annotated.

> *(3) Napari plugin*
>
> *The plugin is well-documented and works by following the installation instructions.*

Great, thanks for the positive feedback.

> *However, I was not able to recreate the segmentations reported in the paper with the default settings for the pre-trained WNet3D: segments are generally too large and there are a lot of false positives. Both the prediction and the final instance segmentation also show substantial border artifacts, possibly due to a block-wise processing scheme.*

Your review here does not match your comments above; above you said it was working well, such that you doubt the GT is real and the data is too easy as it was perfectly easy to threshold with non-learning methods.

You would need to share more details on what you tried. We suggest following our code; namely, we provide the full experimental code and processing for every figure, as was noted in our original submission: https://github.com/C-Achard/cellseg3d-figures.

> ***Reviewer #2 (Public Review):***
>
> *Summary:*
>
> *The authors propose a new method for self-supervised learning of 3d semantic segmentation for fluorescence microscopy. It is based on a WNet architecture (Encoder / Decoder using a UNet for each of these components) that reconstructs the image data after binarization in the bottleneck with a soft n-cuts clustering. They annotate a new dataset for nucleus segmentation in mesoSPIM imaging and train their model on this dataset. They create a napari plugin that provides access to this model and provides additional functionality for training of own models (both supervised and self-supervised), data labeling, and instance segmentation via post-processing of the semantic model predictions. This plugin also provides access to models trained on the contributed dataset in a supervised fashion.*
>
> *Strengths:*
>
> *(1) The idea behind the self-supervised learning loss is interesting.*
>
> *(2) The paper addresses an important challenge. Data annotation is very time-consuming for 3d microscopy data, so a self-supervised method that yields similar results to supervised segmentation would provide massive benefits.*

Thank you for highlighting the strengths of our work and new contributions.

> *Weaknesses:*
>
> *The experiments presented by the authors do not adequately support the claims made in the paper. There are several shortcomings in the design of the experiment and presentation of the results. Further, it is unclear if results of similar quality as reported can be achieved within the GUI by non-expert users.*
>
> *Major weaknesses:*
>
> *(1) The main experiments are conducted on the new mesoSPIM dataset, which contains quite small and well separated nuclei. It is unclear if the good performance of the novel self-supervised learning method compared to CellPose and StarDist would hold for dataset with other characteristics, such as larger nuclei with a more complex morphology or crowded nuclei.*

StarDist is not pretrained, we trained it from scratch as we did for WNet3D. We retrained Cellpose and reported the results both with their pretrained model and our best-retrained model. This is documented in Figure 1 and Suppl. Figure 1. We also want to push back and say that they both work very well on this data. **In fact, our main claim is not that we beat them, it is that we can match them with a self-supervised method.**

> *Further, additional preprocessing of the mesoSPIM images may improve results for StarDist and CellPose (see the first point in minor weaknesses). Note: having a method*

> *that works better for small nuclei would be an important contribution. But I am uncertain the claims hold for larger and/or more crowded nuclei as the current version of the paper implies.*

Figure 2 benchmarks our method on larger and denser nuclei, but we do not intend to claim this is a universal tool. It was specifically designed for light-sheet (brain) data, and we have adjusted the title to be more clear. But we also show in Figure 2 it works well on more dense and noisy samples, hinting that it could be a promising approach. But we agree, as-is, it's unlikely to be good for extremely dense samples like in electron microscopy, which we never claim it would be.

With regards to preprocessing, we respectfully disagree. We trained StarDist (and asked the main developer of StarDist, Martin Weigert, to check our work and he is acknowledged in the paper) and it does very well. Cellpose we also retrained and optimized and we show it works as-well-as leading transformer and CNN-based approaches. Again, we only claimed we can be as good as these methods with an unsupervised approach.

> *The contribution of the paper would be stronger if a comparison with StarDist / CellPose was also done on the additional datasets from Figure 2.*

We appreciate that more datasets would be ideal, but we always feel it's best for the authors of tools to benchmark their own tools on data. We only compared others in Figure 1 to the new dataset we provide so people get a sense of the quality of the data too; there we did extensive searches for best parameters for those tools. So while we think it would be nice, we will leave it to those authors to be most fair. We also narrowed the scope of our claims to mesoSPIM data (added light-sheet to the title), which none of the other examples in Figure 2 are.

> *(2) The experimental setup for the additional datasets seems to be unrealistic. In general, the description of these experiments is quite short and so the exact strategy is unclear from the text. However, you write the following: "The channel containing the foreground was then thresholded and the Voronoi-Otsu algorithm used to generate instance labels (for Platynereis data), with hyperparameters based on the Dice metric with the ground truth." I.e., the hyperparameters for the post-processing are found based on the ground truth. From the description it is unclear whether this is done a) on the part of the data that is then also used to compute metrics or b) on a separate validation split that is not used to compute metrics. If a): this is not a valid experimental setup and amounts to training on your test set. If b): this is ok from an experimental point of view, but likely still significantly overestimates the quality of predictions that can be achieved by manual tuning of these hyperparameters by a user that is not themselves a developer of this plugin or an absolute expert in classical image analysis, see also 3. Note that the paper provides notebooks to reproduce the experimental results. This is very laudable, but I believe that a more extended description of the experiments in the text would still be very helpful to understand the set-up for the reader. Further, from inspection of these notebooks it becomes clear that hyper-parameters where indeed found on the testset (a), so the results are not valid in the current form.*

We apologize for this confusion; we have now expanded the methods to clarify the setup is now **b**; you can see what we exactly did as well in the figure notebook: https://c-achard.github.io/cellseg3d-figures/fig2-b-c-extra-datasets/self-supervised-ext ra.html#threshold-predictions. For clarity, we additionally link each individual notebook now in the Methods.

> *(3) I cannot obtain similar results to the ones reported in the manuscript using the plugin. I tried to obtain some of the results from the paper qualitatively: First I*

> *downloaded one of the volumes from the mesoSPIM dataset (c5image) and applied the WNet3D to it. The prediction looks ok, however the value range is quite narrow (Average BG intensity ~0.4, FG intensity 0.6-0.7). I try to apply the instance segmentation using "Convert to instance labels" from "Utilities". Using "Voronoi-Otsu" does not work due to an error in pyClesperanto ("clGetPlatformIDs failed: PLATFORM_NOT_FOUND_KHR"). Segmentation via "Connected Components" and "Watershed" requires extensive manual tuning to get a somewhat decent result, which is still far from perfect.*

We are sorry to hear of the installation issue; pyClesperanto is a dependency that would be required to reproduce the images (sounds like you had this issue; https://forum.image.sc/t/pyclesperanto-prototype-doesnt-work/45724) We added to our docs now explicitly the fix: https://github.com/AdaptiveMotorControlLab/CellSeg3D/pull/90. We recommend checking the reproduction notebooks (which were linked in initial submission): https://c-achard.github.io/cellseg3d-figures/intro.html.

> *Then I tried to obtain the results for the Mouse Skull Nuclei Dataset from EmbedSeg. The results look like a denoised version of the input image, not a semantic segmentation. I was skeptical from the beginning that the method would transfer without retraining, due to the very different morphology of nuclei (much larger and elongated). None of the available segmentation methods yield a good result, the best I can achieve is a strong over-segmentation with watersheds.*

- We are surprised to hear this; did you follow the following notebook which directly produces the steps to create this figure? (This was linked in preprint): https://c-achard.github.io/cellseg3d-figures/fig2-c-extra-datasets/self-supervised-extra .html

- We have made a video demo for you such that any step that might be unclear is also more clear to a user: (https://youtu.be/U2a9IbiO7nE).

- We also expanded the methods to include the exact values from the notebook into the text.

> *Minor weaknesses:*
>
> *(1) CellPose can work better if images are resized so that the median object size in new images matches the training data. For CellPose the cyto2 model should do this automatically. It would be important to report if this was done, and if not would be advisable to check if this can improve results.*

We reported this value in Figure 1 and found it to work poorly, that is why we retrained Cellpose and found good performance results (also reported in Figure 1). Resizing GB to TB volumes for mesoSPIM data is otherwise not practical, so simply retraining seems the preferable option, which is what we did.

> *(2) It is a bit confusing that F1-Score and Dice Score are used interchangeably to evaluate results. The dice score only evaluates semantic predictions, whereas F1-Score evaluates the actual instance segmentation results. I would advise to only use F1-Score, which is the more appropriate metric. For Figure 1f either the mean F1 score over thresholds or F1 @ 0.5 could be reported. Furthermore, I would advise adopting the recommendations on metric reporting from https://www.nature.com/articles/s41592-023-01942-8.*

We are using the common metrics in the field for instance and semantic segmentation, and report them in the methods. In Figure 2f we actually report the “Dice” as defined in StarDist (as we stated in the Methods). Note, their implementation is functionally equivalent to F1-Score of an IoU >= 0, so we simply changed this label in the figure now for clarity. We agree this clarifies for the expert readers what was done, and we expanded the methods to be more clear about metrics. We added a link to the paper you mention as well.

> *(3) A more conceptual limitation is that the (self-supervised) method is limited to intensity-based segmentation, and so will not be able to work for cases where structures cannot be distinguished based on intensity only. It is further unclear how well it can separate crowded nuclei. While some object separation can be achieved by morphological operations this is generally limited for crowded segmentation tasks and the main motivation behind the segmentation objective used in StarDist, CellPose, and other instance segmentation methods. This limitation is only superficially acknowledged in "Note that WNet3D uses brightness to detect objects [...]" but should be discussed in more depth.*
>
> *Note: this limitation does not mean at all that the underlying contribution is not significant, but I think it is important to address this in more detail so that potential users know where the method is applicable and where it isn't.*

We agree, and we added a new section specifically on limitations. Thanks for raising this good point. Thus, while self-supervision comes at the saving of hundreds of manual labor, it comes at the cost of more limited regimes it can work on. Hence why we don’t claim this should replace excellent methods like Cellpose or Stardist, but rather complement them and can be used on mesoSPIM samples, as we show here.

https://doi.org/10.7554/eLife.99848.1.sa3