Point-based Weakly Supervised 2.5D Cell Segmentation

Fabian Schmeisser[1,2][0000−0001−8222−7900], Andreas Dengel[1,2][0000−0002−6100−8255], and Sheraz Ahmed[1][0000−0002−4239−6520]

[1] German Research Center for Artificial Intelligence (DFKI) GmbH, Kaiserslautern 67663, Germany
[2] RPTU Kaiserslautern-Landau, Kaiserslautern 67663, Germany

Abstract. Volumetric microscopic images show cells in their natural state and solve various problems inherent to 2D projections. The development of competent Deep Learning methods to segment cells in 3D images is, however, held back by the extremely time-consuming and error-prone process of manual ground truth creation. To reduce the burden of manual annotation in 3D, we propose a weakly supervised 2.5D cell segmentation approach that learns to accurately predict 3D segmentation masks from weak, slice-wise point labels. We show that even a single point per cell as ground truth label is sufficient to train a network on par with a fully supervised model that outperforms a top contender of the ISBI Cell Tracking Challenge, and with performance close to that of a fully 3D approach while requiring only a fraction of the resources. The slice-wise, point-based annotation scheme, not only reduces the time required to annotate 3D cell datasets by an estimated factor of 6, but also simplifies the complex and error-prone process of manually segmenting cells using 3D software.

Keywords: Cell Segmentation · Weak Supervision · 2.5D

1 Introduction

Cell segmentation is a cornerstone in biological and pharmaceutical research. Accurately segmented cells in microscopy images help with the development of novel treatments for a wide variety of diseases. As such, cell segmentation paves the way to diagnose deadly diseases like various forms of cancer early, or to prevent them altogether. Especially 3D images, acquired with modern imaging methods like Z-stack acquisition, show individual cells in a more life-like state and can resolve problems that are inherent to 2D projections. With 3D images of cell cultures, overlapping cells can be clearly separated, and rotation or movement in z-direction cannot be mistaken for a shape change of a cell. A massive hurdle to overcome in this domain is the extensive labor and the necessity of expert knowledge when manually segmenting cells in microscopic images. This problem is especially prevalent for three-dimensional image stacks of cell cultures, where manual segmentation necessitates the individual annotation of single slices or

the complex and error-prone annotation via 3D visualization software. With the rise of computer vision methods and specifically deep learning (DL) over the past decade, many approaches have been developed that ease this burden by employing computer-aided systems to quickly and accurately segment cells [16] [6] [15]. Given that the nature of these technologies is, however, often reliant on fully supervised DL methods, the training of such systems is inherently intertwined with the manual creation of ground truth masks. In this study, we present a novel 2.5D cell segmentation strategy that leverages weakly annotated ground truth to predict full segmentation masks with accuracy effectively equal to full supervision. By adapting point-based supervision [4] which has been proven to be successful in 2D cell segmentation [11], leveraging different 2.5D data augmentation techniques to encode spatial information in the Z-dimension into 2D image slices, and integrating an intersection-based slice stacking method, the method saves computational resources during training and inference in addition to reducing the time required for the manual annotation of cells.

We show that our method outperforms one of the top-ranked methods submitted to the ISBI Cell Tracking Challenge [13] in both, fully and weakly supervised settings, and comes close in performance compared to a popular 3D approach [19] while requiring only a fraction of the resources. We additionally provide insight into the influence of data pre-processing strategies used to encode z-directional spatial information into 2D slice representations, as well as post-processing strategies used to re-construct 3D masks from slice-wise predictions.

2 Related Work

Our point-supervision 2.5D cell segmentation method rests on the two foundational pillars of 2.5D/3D instance segmentation and weak supervision using weak labels. As, to the best of our knowledge, a combination of these two fundamentals does not exist in published literature on the topic, we discuss each pillar individually in this section. Fundamentally, image analysis methods that deal with 3D images can be classified into two general categories: 2.5D and 3D. For a very general distinction, 3D methods deal with fully volumetric in- and output, while 2.5D methods take 2D slices of a volumetric image as input [24]. In general, 2.5D methods are much less intensive on resources and can thus run even on lower-end hardware not necessarily specialized for deep learning tasks.

3D Cell Segmentation Weigert et al. [19] present a 3D cell instance segmentation method based on a modified 3D ResNet neural network backbone to predict star convex polyhedra representations and values indicating the likelihood of a pixel being part of an object. This fully volumetric approach is tested on isotropic data and relies on convex, elliptic cell shapes to produce accurate segmentation masks. Jelli et al. [9] improve upon the aforementioned method with a novel post-processing algorithm that allows for accurate segmentation of cells even if they are not necessarily elliptic. Eschweiler et al. [7] leverage the predictive capabilities of the 3D U-Net network architecture [5] coupled with Watershed post-processing to generate segmentation masks from a volumetric

input. Just minor modifications to the number of filters of 3D U-Net are shown to yield impressive results for 3D cell segmentation.

2.5D Cell Segmentation Cellpose [16] on the other hand, is a generalist approach primarily developed for 2D cell segmentation, that is extended to 3D by predicting masks on each slice in x, y, and z direction and fusing the predictions together for a final 3D segmentation output. Wagner and Rohr [18] introduce a novel data augmentation strategy to integrate information about neighboring slices in z-direction into a 2D input for their proposed cell segmentation pipeline. This data augmentation strategy called *pseudocoloring* is based on a rough, non-DL, pre-segmentation of neighboring slices and subsequent assembly of three neighboring slices to mimic a typical RGB image input. Wu et al. [22] suggest a strategy based on ensemble learning and slice fusion for three-dimensional nuclei instance segmentation. The authors use an ensemble of various Mask R-CNN adaptations, a network architecture commonly used in 2D cell segmentation tasks [6]. Nuclei segmentation masks for each 2D slice of a volume are generated by using several object detectors and then merged into a 3D segmentation mask. Finally, Scherr et al. [13] employ an adapted U-Net with two decoder paths to predict neighbor distances of individual cells. The authors achieved multiple top-3 rankings, which improved versions of the approach still occupy, in the ISBI Cell Tracking Challenge [17].

Weakly Supervised Cell Segmentation In the realm of weak supervision, we differentiate between two distinct forms: *Missing or incomplete annotations*, where networks are trained on datasets containing inaccurate or incomplete ground truth masks, and *weak labels*, where datasets are densely labeled, but labels are of lower information density and cheaper to produce than full mask annotations. While various popular approaches in 3D cell segmentation deal with the first form of weak supervision, missing or incomplete annotations, by ignoring loss calculation at these image regions [1] [25], or by leveraging synthetic data to strengthen poor annotation quantity [26] [23], the usage of weak labels has, to the best of our knowledge, not yet been applied to 3D cell segmentation. In the realm of 2D cell segmentation, segmentation techniques employing weak labels are more prevalent. Point2Mask [11] is based on an augmented version of the Mask R-CNN architecture, that uses sparse point labels instead of full ground truth masks. Based on [4], ground truth annotations consist of automatically generated, randomly sampled points inside a manually annotated bounding box containing a single cell instance. The authors show, that even though the time taken to annotate cell instances is reduced by up to 6 times, prediction accuracy of the network is close to that of a fully supervised method.

3 Dataset

The severe lack of accurately and fully annotated ground truth data is one of the biggest issues holding back the development of reliable and generalizable 3D cell segmentation methods, and the main motivation for the development of our approach. To fairly judge the performance of our approach against fully supervised

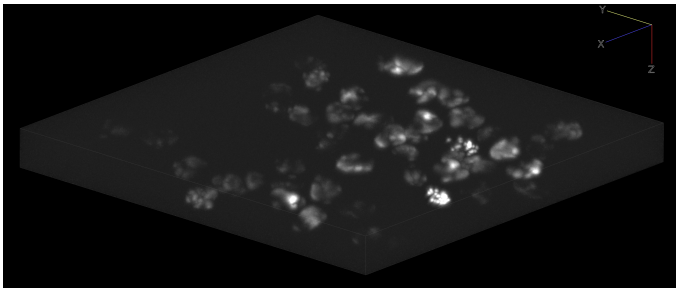

Fig. 1. Sample volume of the N3DH-SIM+ dataset

methods, we decided on a synthetic dataset provided by the ISBI Cell Tracking Challenge [17] website, which is by design guaranteed to have dense, highly accurate ground truth segmentation masks. The dataset *Fluo-N3DH-SIM+*, as shown in figure 1, contains 230 anisotropic 3D images of simulated C.elegans cells with resolutions ranging from $59x639x349$ to $59x652x642$. We henceforth refer to the respective dimensions of a volume as x, y, and z, where z describes the first, x the second, and y the third value (i.e. $ZxXxY$) of a volume's resolution. As the dataset consists of two distinct time sequences, comprised of 150 and 80 volumes respectively, we manually choose training, validation and test splits. We split the dataset into training data (170 images), validation data (20 images) and testing data (40 images). The first 120/50 images of time sequences 1/2 are used for training, the next 10/10 for validation, and the final 20/20 for testing. With this dataset partitioning, we can ensure that our network is capable of generalizing from initial information of a sequence to unseen information at later time steps.

4 Proposed Method

Our proposed method is composed of various building blocks, that are described in the sections below. The pre- and post-processing schemes, as well as supervision modality, can be interchanged to find the best-working overall strategy for a particular dataset.

4.1 2.5D Data Augmentation

To test how spatial context between neighboring slices can be leveraged for 2.5D training, we use three different augmentation techniques to prepare 2D image slices taken from the 3D images for processing by our network.

Single Slice Input. The input is composed of only a single grayscale image slice in z-direction and its respective ground truth segmentation masks. With this input modality, no spatial context associated with other objects in the z-direction is encoded.

Three-Slice Input. The input consists of one slice s alongside the corresponding ground truth, as well as the slices directly neighboring s in z-direction, $s-1$ and $s+1$. The slices are concatenated along the channel axis, and the final input resembles a typical RGB-image.

Context-aware Pseudo Coloring. We adapt the context aware pseudo coloring pre-processing method proposed in [18]. The method consists of several pre-processing steps for each three-slice stack, that aims to highlight regions in the center slice where cells are located in neighboring slices. With the use of contrast limited adaptive histogram equalization filters, a rough pre-segmentation via thresholding and multiplication of intensity values, a pseudo-colored image of the central slice is generated that supports the model's capabilities of capturing spatial context.

4.2 Model

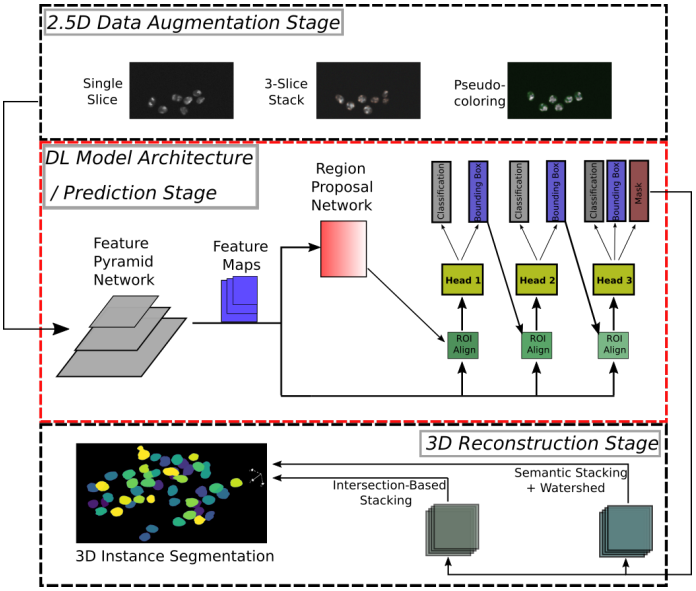

Fig. 2. Proposed pipeline for weakly supervised 2.5D cell segmentation. The structural overview of the DL model architecture is outlined in red. For each data augmentation strategy, individual models are trained. The reconstruction strategies are model agnostic.

We adapt the Cascade Mask-RCNN architecture [3] with a ResNet50 Feature Pyramid Network backbone [12], which has proven to be a successful tool for cell segmentation in 2D and excels at learning from weak labels [10]. The proven viability of this network in combination with point supervision makes it an ideal

candidate to introduce point supervision to the realm of 3D cell segmentation. The three main building blocks of the pipeline as shown in figure 2 are as follows:
Feature Pyramid Network (FPN) with ResNet50 Backbone as the first block of the pipeline has the purpose of extracting feature maps from the input image at varying scales. The bottom-up and top-down pathways with lateral connections allow the network to extract high-resolution, semantically strong, as well as low-resolution, semantically weak features.
Region Proposal Network (RPN) receives the extracted feature maps from the FPN backbone and fulfills the task of detecting Regions of Interest (RoIs) and aligning them with the ground truth.
The Prediction Heads finally, have the purpose of outputting the segmentation masks, as well as bounding boxes and object classes. The 3-stage cascading mask heads further improve segmentation performance over the singular mask head of a standard Mask RCNN [8]. This improvement is achieved by increasing IoU thresholds for each prediction head and thus refining segmentation predictions by providing more accurate bounding boxes to the mask head. We adapt the loss calculation according to [4] for weakly supervised training.

4.3 Pointly Supervision

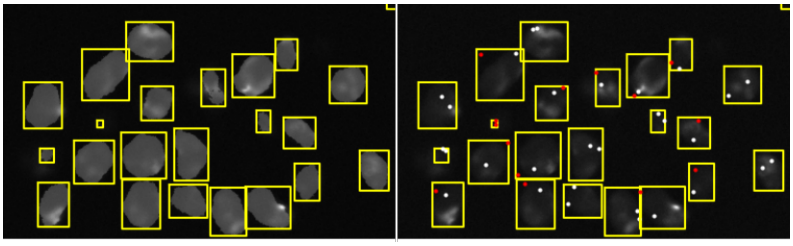

Fig. 3. Comparison of ground truth annotations overlayed over the corresponding image slice. Full mask annotation (left) and point annotation (right). Points inside cells are marked in white, points outside cells in red. Bounding boxes surrounding the cells in yellow.

For Pointly Supervision, the annotation approach of object instances is as follows: A bounding box is manually drawn by an annotator around each object instance in the dataset. Within this bounding box, a set number of points is generated at random positions. The annotator now has to decide if the point lies inside an object or outside the object and marks the point as 1 or 0 respectively. Depending on the number of points generated, this leads to an estimated annotation speed-up of between 2 and 6 times [11]. The model's mask head loss calculation is adapted to calculate a loss for the point annotations by employing bilinear interpolation to approximate the predictions at the location of ground truth points instead of comparing its output to full ground truth masks. The

same loss function that is used for full supervision can be applied to the point labels in this way and the error is backpropagated through the interpolation in addition to the pipeline architecture.

4.4 Post-Processing

We employ two distinct post-processing schemes to re-construct 3D segmentations from the slice-wise predictions produced by our pipeline.

Semantic Reconstruction and Watershed Post-Processing. The first post-processing scheme discards the instantiated outputs and instead fuses the predictions only on a fore- and background basis. This results in a 3D semantic segmentation mask that is subsequently instantiated using the 3D Watershed Algorithm [2]. Due to a lower likelihood of overlapping cells in microscopic images with sufficiently high resolution along the Z-axis, the Watershed Algorithm is expected to perform well on the semantic 3D masks. This is evidenced by the prevalence of creating semantic-level segmentation predictions using a deep learning algorithms and instantiation via Watershed in 3D cell segmentation [7] [18] [20].

Intersection-Based Slice Stacking. The second scheme directly leverages the instantiated output produced by the ROI-based method and accumulates predictions through the Z-axis by clustering instance predictions with high overlap into a single 3D instance. A visual representation of this clustering scheme can be

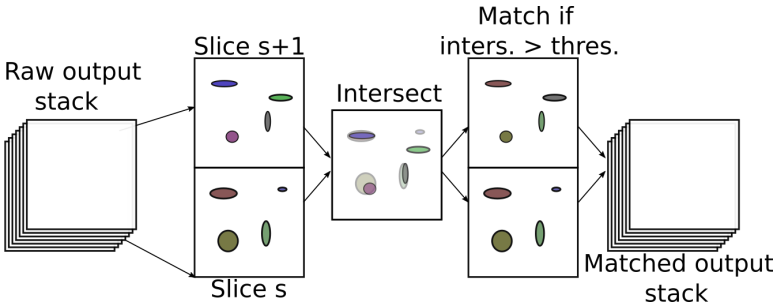

Fig. 4. Intersection-based object matching. The intersection score is calculated as overlap between two objects divided by total area of the smaller object.

seen in figure 4. Since all objects in the predicted image are already instantiated by using an R-CNN RoI detection scheme, this method is less prone to split singular objects into two, due to poorly generated seed points, or ambiguous distance transforms. This is a problem often encountered when using the Watershed Algorithm without sufficient preliminary knowledge of object structure in the dataset, and thus incorrect choice of parameters. We therefore assume that this reconstruction method is more generalizable when analyzing unknown data.

4.5 Metrics

We choose two different metrics to evaluate quantitative performance of our approach and baseline methods, namely SEG and Accuracy. The SEG metric as defined in [17] is based on the Jaccard similarity index (often also referred to as Intersection over Union (IoU)) which measures the similarity between two sets of pixels:

$$IoU(GT, P) = \frac{|GT \cap P|}{|GT \cup P|} \tag{1}$$

where GT is the set of pixels describing a ground truth mask, and P is the set of pixels that form a prediction. To match a ground truth instance with a prediction instance, the following condition must be true:

$$|GT \cap P| > 0.5 \cdot |GT| \tag{2}$$

For each GT object, at most one predicted object can be considered matching. If there is no predicted object that satisfies the above condition, the score for this GT object is set to 0. Otherwise, the score is set to the corresponding IoU. The SEG score is then calculated as a mean over all measured scores for each GT object.
The SEG metric, however, does not take False Positives into account since predictions that do not match a GT object are not considered. We therefore additionally use the Accuracy metric

$$Acc(GT, P) = \frac{TP}{TP + FP + FN} \tag{3}$$

which takes true positives (TP), false positives (FP) and false negatives (FN) into account. We use different IoU thresholds to count the number of predictions that are considered TPs, FPs, and FNs in the range of $[0.1, 0.2, \cdots, 0.9]$. For the Accuracy at a certain IoU threshold X, we use the abbreviation Acc@X.

5 Results

To show the validity of our segmentation approach, we first draw a quantitative comparison between our method, a 2.5D method that is one of the top contenders of the ISBI Cell Tracking Challenge, and a fully volumetric method. All algorithms are trained and tested on identical subsets of the Fluo-N3DH-SIM+ dataset. We continue by evaluating performance respective to the employed 2.5D pre-processing scheme, and compare weakly supervised performance with respect to the number of points used for ground truth annotation.

5.1 Comparison to SOTA

To show the validity of the pipeline architecture, we compare its performance to the algorithm proposed by [14] (KIT-SCHE) and the latest version of the popular volumetric approach Stardist3D [19] with a 3D U-Net backbone. KIT-SCHE

currently occupies top-3 spots in the Cell Segmentation Benchmark leaderboard for nine different datasets, including three 1st places. The official implementations of KIT-SCHE [14] and Stardist3D are used for training and evaluation. KIT-SCHE is trained for 200 epochs, Stardist3D is trained for 2000 epochs. We use the same train/test/validation splits as for the training of our method. For evaluation, we employ the official evaluation software published by the organizers of the Cell Tracking Challenge, which calculates prediction quality using the SEG metric [17]. Note that the results for KIT-SCHE provided in this study may differ from the results posted on the Cell Tracking Challenge leaderboard. Our test split differs from the official test set used to calculate results for the leaderboard, as the latter does not come with publicly available ground truth. The results of the quantitative comparison can be seen in table 5.1.

Table 1. Comparison between the fully supervised setup of our method against KIT-SCHE [14] and Stardist3D [19]. Our method outperfoms the 2.5D method KIT-SCHE in all setups and has a SEG score close to the fully volumetric Stardist method. VRAM usage is measured as the minimum required amount for full image resolution, using a batch size of 1.

Method	SEG metric	Acc @ 50	Acc @ 70	VRAM usage	Modality
KIT-SCHE	0.639	0.580	0.341	3.18 GB	2.5D
Ours (1-slice)	0.666	0.771	0.438	2.73 GB	2.5D
Ours (3-slice)	0.732	0.848	0.581	2.90 GB	2.5D
Ours (pseudocoloring)	0.646	0.713	0.415	2.74 GB	2.5D
Stardist3D	0.793	0.951	0.875	43.80 GB	3D

Our approach outperforms KIT-SCHE in all setups, but specifically, our best setup, 3-slice training, outperforms it by almost 0.1 in the SEG metric. The difference in performance becomes even more apparent with the accuracy metric, where even our weakest setup has a significant advantage over KIT-SCHE at both, 50% and 70% IoU thresholds. The large difference in accuracy is most likely due to more false positive predictions by KIT-SCHE, which are not penalized using the SEG metric. Qualitative results show a similar pattern. Figure 5 provides a comparison between ground truth, KIT-SCHE, and our method in 3D and single slice views.

The comparison shows the tendency of our approach to under-segment small objects, as well as avoidance of false positives. This proves to be beneficial to prediction accuracy, as inaccurate segmentation of less visible objects in 2D as well as over-segmentation lead to less accurate results in 3D. When objects are detected correctly by both algorithms, our approach shows better area coverage and shape approximation.

In comparison to Stardist3D, our approach slightly underperforms in SEG and Acc@50 measures. However, during training Stardist3D requires over 40 GB of VRAM, which is only provided by specialized deep learning hardware. Additionally, Stardist3D can only be trained with fully and accurately annotated 3D

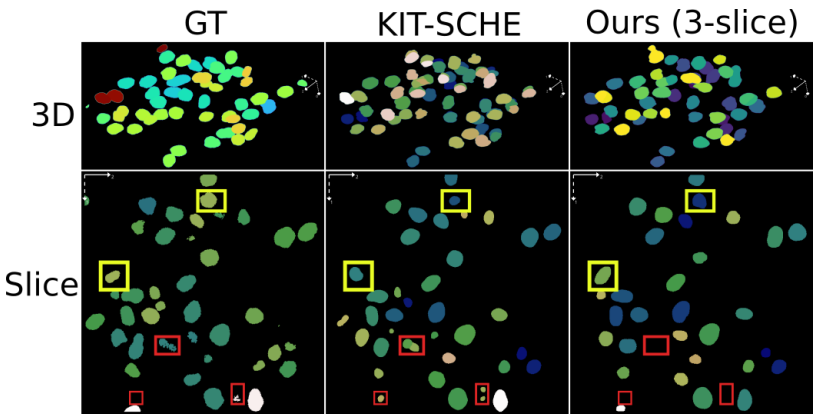

Fig. 5. Comparison between our approach in a fully supervised setup against KIT-SCHE. The red boxes show examples of over-segmentation or segmentation of barely visible objects, which are detrimental to accuracy in 3D reconstructions. Yellow boxes show better cell shape approximation and area coverage by our approach of cells correctly identified by both approaches.

ground truth masks, while even the fully supervised iteration of our approach can be trained on partial, slice-wise annotations. In any setting that does not provide high-end hardware and perfect annotations, as is the case for many biological and pharmaceutical research institutes and data, training the fully volumetric approach is not feasible.

5.2 Number of Weak Annotations, 2.5D Pre-Processing and 3D Reconstruction

We assess quantitative performance of different supervision, pre-, and post-processing modalities using 3D performance metrics. The SEG metric employed by the ISBI Cell Tracking Challenge gives a good estimate of overall performance, while the accuracy at different thresholds can provide more detailed insights. Figure 6 shows an overview of SEG scores achieved by our models with respect to post-processing scheme, number of points used for supervision, and 2.5D data pre-processing.

Overall, the 3-slice 2.5D pre-processing scheme far outperforms the other methods in every constellation. Applying pseudo-coloring to the model input matches single-slice input when a low number of points is used for supervision, but shows diminishing returns for higher number of points. While both, 3-slice and pseudocolored inputs, are meant to provide spatial context in z-direction, the pseudocoloring scheme does not encode this spatial information in a way that is significant to our model's performance.

In contrast to [4] and [11] we do not see significant differences in quantitative performance for a reduced number of points in the case of 3-slice input. We attribute this to the comparatively less complex object shapes contained in the

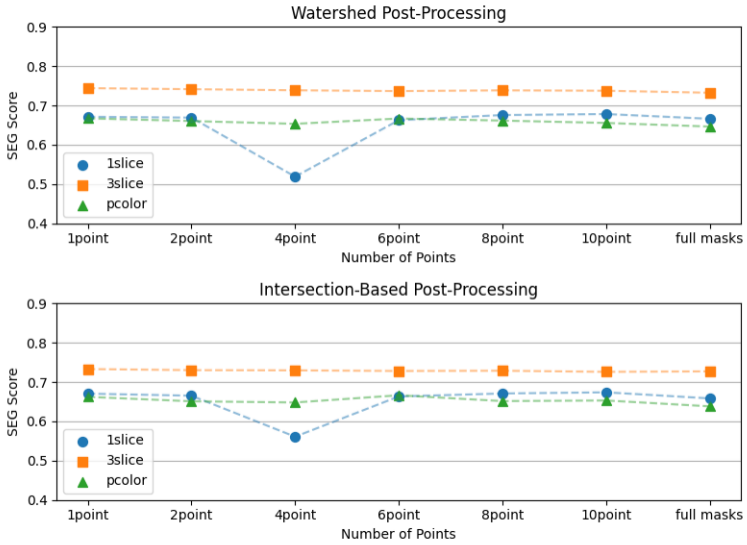

Fig. 6. SEG scores for the different training, post-processing, and 2.5D data augmentation modalities.

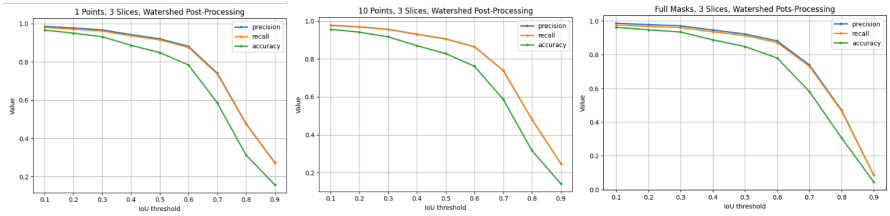

Fig. 7. Accuracy, Recall, and Precision at different IoU thresholds for 1 and 10 point supervision, as well as full supervision.

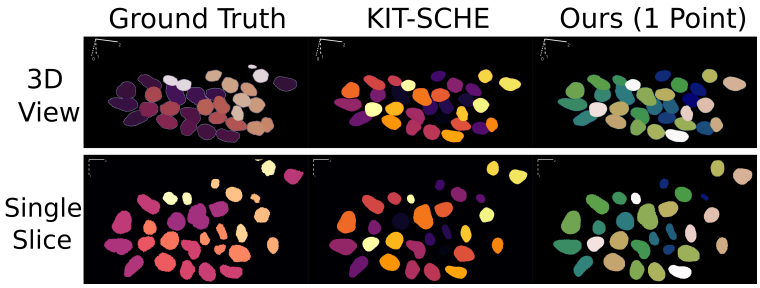

Fig. 8. Comparison between ground truth, KIT-SCHE[14] and our method. Our approach is trained using only 1 point as ground truth label, while KIT-SCHE is trained with full mask annotations. Visualizations are done with the Napari software [21]

dataset and unclear object outlines inherent to slice-wise representations of 3D microscopic images. Since even human expert annotators often cannot match ground truth with IoUs over 80% for 3D cell images [9], the output of 2.5D and 3D segmentation methods has a higher dependency on estimation of object shapes. This leads to our point supervision approach achieving results equal to that of full supervision, even if only a single point is available as ground truth. Concerning the difference between post-processing methods, Watershed post-processing slightly outperforms Intersection-Based slice stacking for stronger setups, while Intersection-Based post-processing elevates quantitative performance for weaker setups.

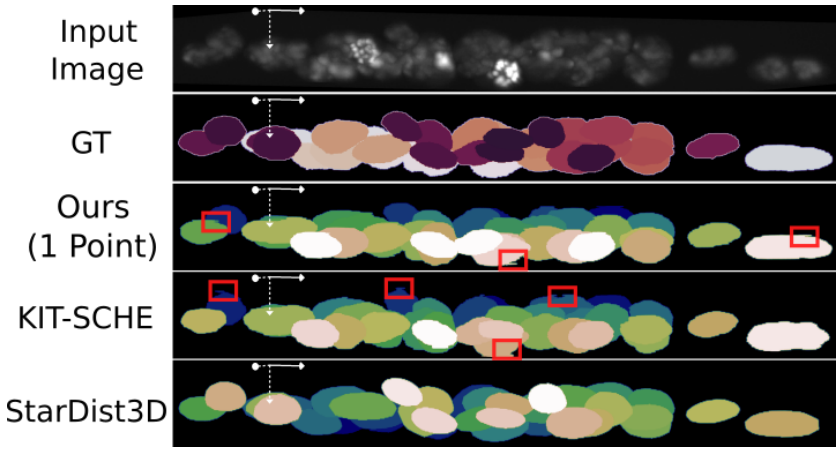

Fig. 9. Side-view of mask predictions by KIT-SCHE, Stardist3D, and our method. The red boxes indicate jagged edges, 3D reconstruction artifacts inherent to 2.5D methods. Full depth context allows volumetric methods to produce smoother segmentation masks.

A comparison of the accuracy, precision, and recall metrics for the best performing setup, 3-slice input and watershed-aided reconstruction, using 1 point, 10 point, or full mask supervision, can be seen in figure 7. We have near equivalent performance at all IoU thresholds, regardless of the number of points used for supervision. We can therefore assume that single point supervision is completely sufficient to train a segmentation model on the dataset Fluo-N3DH-SIM+.
The most notable qualitative difference between the 2.5D setups and the fully volumetric Stardist3D can be seen in figure 9. Due to the necessity of slice-stacking to reconstruct 3D predictions, KIT-SCHE and our approach show notable edges. While Stardist3D produces smoother results, this can sometimes be detrimental to accurate cell coverage, as exemplified by the rightmost large cell instance in figure 9.

5.3 Estimated Time Savings When Using Weak Annotations

We estimate the time saved during the annotation process by consulting two sources, namely [9] and [11], that provide approximate times for annotating cells in 3D and in 2D respectively. Expert annotation of a single cell in 3D requires between 300 and 420 seconds [9]. The authors find, however, that such annotations are often of low quality and in need of refinement. We assume a similar correlation between full mask and bounding box annotation speed in 3D as in 2D, given as $11x$ [11], and equal time for 3D mask refinement and refinement of the 3D bounding box to more accurately enclose objects when viewed slice-wise. Then, bounding box generation requires between 27 and 38 seconds for one cell. With cells spanning an average of 28 Z-slices in the Fluo-N3DH-SIM+ dataset and an average time of 0.9 seconds per point annotation as in [11], the total annotation time for a single cell using single point annotation and bounding boxes is approximately 51.3 to 62.3 seconds. Overall, we can thus expect a speed-up of roughly 6 times when comparing a full 3D mask annotation to our proposed point annotation scheme.

6 Conclusion

We propose a 2.5D cell segmentation approach that outperforms one of the highest-ranking approaches on the ISBI Cell Tracking Challenge leaderboard by a fair margin. Moreover, the proposed approach can be trained with only a single point as ground truth, without diminishing its quantitative and qualitative performance. By benchmarking and evaluating different data pre-processing strategies as well as 3D reconstruction strategies, we find that our algorithm significantly benefits from the spatial context provided by 3-slice input. The annotation process for 3D microscopic images is known to be not only extremely time-consuming, but also highly prone to errors. Using single point-based weak labels as ground truth for a 2.5D deep learning algorithm makes the annotation of full 3D datasets feasible. Especially in cases of unclear boundaries and barely visible objects, problems that are currently unavoidable for microscopic images acquired through Z-stack acquisition, slice-wise point annotation can even eliminate biases introduced through approximations of cell shapes. Our method therefore provides a way to efficiently prepare the vast amount of unlabeled 3D microscopic image data that is available through various sources for deep learning-based segmentation methods. By reducing this burden to a manageable degree, our work can thus support researchers in studying cells in a more life-like 3D representation far more effectively.

References

[1] Assaf Arbelle, Shaked Cohen, and Tammy Riklin Raviv. “Dual-Task ConvLSTM-UNet for Instance Segmentation of Weakly Annotated Microscopy Videos”. In: *IEEE Transactions on Medical Imaging* 41.8 (Aug. 2022), pp. 1948–1960. DOI: 10.1109/tmi.2022.3152927.

[2] Serge Beucher and Fernand Meyer. "The morphological approach to segmentation: the watershed transformation". In: *Mathematical morphology in image processing*. CRC Press, 2018, pp. 433–481.

[3] Zhaowei Cai and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6154–6162.

[4] Bowen Cheng, Omkar Parkhi, and Alexander Kirillov. "Pointly-supervised instance segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 2617–2626.

[5] Özgün Çiçek et al. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2016, pp. 424–432. ISBN: 9783319467238. DOI: 10.1007/978-3-319-46723-8_49.

[6] Christoffer Edlund et al. "LIVECell—A large-scale dataset for label-free live cell segmentation". In: *Nature Methods* 18.9 (Aug. 2021), pp. 1038–1045. DOI: 10.1038/s41592-021-01249-6.

[7] Dennis Eschweiler, Richard S. Smith, and Johannes Stegmaier. "Robust 3d Cell Segmentation: Extending The View Of Cellpose". In: *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, Oct. 2022. DOI: 10.1109/icip46576.2022.9897942.

[8] Kaiming He et al. "Mask R-CNN". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Feb. 2020), pp. 386–397. DOI: 10.1109/tpami.2018.2844175.

[9] Eric Jelli et al. "Single-cell segmentation in bacterial biofilms with an optimized deep learning method enables tracking of cell lineages and measurements of growth rates". In: *Molecular Microbiology* 119.6 (Apr. 2023), pp. 659–676. ISSN: 1365-2958. DOI: 10.1111/mmi.15064.

[10] Nabeel Khalid et al. "PACE: Point Annotation-Based Cell Segmentation for Efficient Microscopic Image Analysis". In: *International Conference on Artificial Neural Networks*. Springer. 2023, pp. 545–557.

[11] Nabeel Khalid et al. "Point2Mask: A Weakly Supervised Approach for Cell Segmentation Using Point Annotation". In: *Medical Image Understanding and Analysis*. Springer International Publishing, 2022, pp. 139–153. DOI: 10.1007/978-3-031-12053-4_11.

[12] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[13] Tim Scherr et al. "Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy". In: *PLOS ONE* 15.12 (Dec. 2020). Ed. by Konradin Metze, e0243219. DOI: 10.1371/journal.pone.0243219.

[14] Tim Scherr et al. "On improving an already competitive segmentation algorithm for the Cell Tracking Challenge-lessons learned". In: *bioRxiv* (2021), pp. 2021–06.

[15] Uwe Schmidt et al. "Cell Detection with Star-convex Polygons". In: (2018). DOI: 10.48550/ARXIV.1806.03535.

[16] Carsen Stringer et al. "Cellpose: a generalist algorithm for cellular segmentation". In: *Nature Methods* 18.1 (Dec. 2020), pp. 100–106. DOI: 10.1038/s41592-020-01018-x.

[17] Vladimír Ulman et al. "An objective comparison of cell-tracking algorithms". In: *Nature Methods* 14.12 (Oct. 2017), pp. 1141–1152. DOI: 10.1038/nmeth.4473.

[18] Royden Wagner and Karl Rohr. *EfficientCellSeg: Efficient Volumetric Cell Segmentation Using Context Aware Pseudocoloring*. 2022. DOI: 10.48550/ARXIV.2204.03014.

[19] Martin Weigert et al. "Star-convex polyhedra for 3D object detection and segmentation in microscopy". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2020, pp. 3666–3673.

[20] Adrian Wolny et al. "Accurate and versatile 3D segmentation of plant tissues at cellular resolution". In: *eLife* 9 (July 2020). DOI: 10.7554/elife.57613.

[21] How Does napari Work. "napari: A Multidimensional Image Viewer for Python A guide to some of the key concepts of napari". In: ().

[22] Liming Wu et al. "An Ensemble Learning and Slice Fusion Strategy for Three-Dimensional Nuclei Instance Segmentation". In: (Apr. 2022). DOI: 10.1101/2022.04.28.489938.

[23] Liming Wu et al. "NISNet3D: Three-Dimensional Nuclear Synthesis and Instance Segmentation for Fluorescence Microscopy Images". In: (June 2023). DOI: 10.1101/2022.06.10.495713.

[24] Yichi Zhang et al. "Bridging 2D and 3D segmentation networks for computation-efficient volumetric medical image segmentation: An empirical study of 2.5D solutions". In: *Computerized Medical Imaging and Graphics* 99 (July 2022), p. 102088. DOI: 10.1016/j.compmedimag.2022.102088.

[25] Zhuo Zhao et al. "Deep Learning Based Instance Segmentation in 3D Biomedical Images Using Weak Annotation". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2018, pp. 352–360. ISBN: 9783030009373. DOI: 10.1007/978-3-030-00937-3_41.

[26] Amirkoushyar Ziabari et al. *YOLO2U-Net: Detection-Guided 3D Instance Segmentation for Microscopy*. 2022. DOI: 10.48550/ARXIV.2207.06215.