



OPEN

A novel deep learning-based 3D cell segmentation framework for future image-based disease detection

Andong Wang^{1,3}, Qi Zhang^{1,3}, Yang Han¹, Sean Megason², Sahand Hormoz², Kishore R. Mosaliganti², Jacqueline C. K. Lam¹✉ & Victor O. K. Li¹✉

Cell segmentation plays a crucial role in understanding, diagnosing, and treating diseases. Despite the recent success of deep learning-based cell segmentation methods, it remains challenging to accurately segment densely packed cells in 3D cell membrane images. Existing approaches also require fine-tuning multiple manually selected hyperparameters on the new datasets. We develop a deep learning-based 3D cell segmentation pipeline, 3DCellSeg, to address these challenges. Compared to the existing methods, our approach carries the following novelties: (1) a robust two-stage pipeline, requiring only one hyperparameter; (2) a light-weight deep convolutional neural network (3DCellSegNet) to efficiently output voxel-wise masks; (3) a custom loss function (3DCellSeg Loss) to tackle the clumped cell problem; and (4) an efficient touching area-based clustering algorithm (TASCAN) to separate 3D cells from the foreground masks. Cell segmentation experiments conducted on four different cell datasets show that 3DCellSeg outperforms the baseline models on the ATAS (plant), HMS (animal), and LRP (plant) datasets with an overall accuracy of 95.6%, 76.4%, and 74.7%, respectively, while achieving an accuracy comparable to the baselines on the Ovules (plant) dataset with an overall accuracy of 82.2%. Ablation studies show that the individual improvements in accuracy is attributable to 3DCellSegNet, 3DCellSeg Loss, and TASCAN, with the 3DCellSeg demonstrating robustness across different datasets and cell shapes. Our results suggest that 3DCellSeg can serve a powerful biomedical and clinical tool, such as histo-pathological image analysis, for cancer diagnosis and grading.

Computer-aided digital pathology plays an increasingly important role in understanding, diagnosing, and treating various kinds of diseases¹. Digital image processing has become increasingly widespread in biological research as high-throughput/high-content microscopy and screening generate significant quantities of complex fine-grained cellular images². With the development of modern biomedical techniques—such as cellular staining, whole-slide imaging, telemedicine, and cloud storage—millions of tissue biopsies are analysed annually^{3,4}. To increase the efficiency of screening big cell image data, and to standardize image analysis and reduce variation in interpretability, there is a need to develop improved computer vision techniques for big cell image data analysis.

Cell segmentation plays a key role in biological image processing. Computer-aided diagnostics requires the identification of single cells. With regards to histo-pathological image analysis for cancer diagnosis and grading, the regularity of cell borders, shapes, and distributions provides an important insight into whether tissue regions are cancerous⁵. Other recent research studies have shown that the distribution and quality of blood cells are connected with the pathogenesis of Alzheimer's Disease and may contribute to disease progression⁶. Cell segmentation has also been applied in studying the dynamics of gene regulation, cell growth and proliferation. Time-lapse microscopy technologies—including confocal, two photon, and light sheet microscopy—enable detailed data analytics based on dynamic cellular processes at the single-cell level^{7,8}. However, recognizing cells as the objects of an image, and tracking these objects from one image to the next, still presents a central challenge⁸. The development of computational cell segmentation methods dramatically decreases time and labour in related biomedical applications.

Cell segmentation algorithms can be categorized into semantic segmentation and instance segmentation. Semantic segmentation refers to the partitioning of images into different semantic parts and assigning each pixel to a class (e.g. cell foreground or background). Instance segmentation seeks to identify each instance of the

¹Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China. ²Department of Systems Biology, Harvard Medical School, Boston, MA, USA. ³These authors contributed equally: Andong Wang and Qi Zhang ✉email: jcklam@eee.hku.hk; vli@eee.hku.hk

Segmentation type		Deep learning model for 2D cell segmentation	Deep learning model for 3D cell segmentation	Major drawback
Semantic segmentation		U-Net ⁹ ; DeepCell ¹⁰	3D U-net ¹¹ ; V-Net ¹² ; VoxResNet ¹³ ; 3D-DSN ¹⁴ ; 2D-3D ¹⁵ ; C2FNAS ¹⁶ ; Automatic Data Augmentation ¹⁷	Fails to distinguish different cell instances
Instance segmentation	Contour-aware approach	DCAN ¹⁸ ; Deep Watershed ¹⁹	U-Net + CRF ²⁰ ; U-Net + SWS ²¹ ; PlantSeg ²² ; DISCO ²³ ; U-Net + Graph based ²⁴	Performance is highly dependent on the manually selected parameters during the post-processing procedures Prone to fuse cells that are tightly adhered
	Object-detection-based	Retinanet ²⁵ ; R-CNN, and a series of revised structures ^{26–28} ; Keypoint bounding box ²⁹ ; PointINS ³⁰ ; FCOS ³¹ ; CenterMask ³² ; YOLACT ³³	Retina-Unet ³⁴ ; Weak Annotation ³⁵	Suffers from a severe imbalance between the number of positive and negative anchor boxes May fail to discern objects that are poorly approximated with bounding boxes
	Other strategies	GAN ³⁶ ; Embedding ^{37,38} ; StarDist ³⁹ ; TensorMask ⁴⁰ ; AdaptIS ⁴¹ ; CondInst ⁴²	StarDist 3D ⁴³ ; ShapeMetrics ⁴⁴ ; Spherical Harmonics ⁴⁵	Less accurate than the previous two mainstream strategies Many of these models are based on specific assumptions The training process of GAN networks is highly complex, especially on 3D datasets

Table 1. An overview of the existing deep learning models for 2D/3D cell segmentation.

same class, by separately detecting and delineating every single cell shown in the image. Table 1 overviews the current deep learning-based methods for cell segmentation and lists their major drawbacks. For a more detailed description of traditional methods and deep learning-based approaches for cell instance segmentation, please refer to Supplementary Information.

Most existing 3D segmentation deep learning models focus on semantic segmentation. 3D U-Net¹¹ extends 2D U-Net into 3D, incorporating a path for extracting high-level features and a path for generating segmented cells in full-resolution. V-Net is another 3D version of U-Net, with residual connections added between the convolutional layers¹². VoxResNet extends a 2D deep residual network to a 3D residual network¹³. 3D-DSN is a 3D fully convolutional network equipped with a deep supervision mechanism¹⁴. These methods have been applied in various biomedical segmentation tasks, including image segmentation for brain, liver, prostate, and heart tissues. C2FNAS (coarse-to-fine neural architecture search)¹⁶ automatically identifies a 3D segmentation network, and¹⁷ employs automatic data augmentation for medical image segmentation tasks. Both of these two models^{16,17} have achieved state-of-the-art performance.

Some recent methods have applied instance segmentation on 3D cellular images^{21–23,25,40,41}. These methods use deep learning based 3D semantic segmentation models in their first step to generate a pixel-based classification of the cell interiors, edges, and backgrounds. Traditional techniques, including thresholding and watershed, are then used to separate single cells from each other. Their major drawback is in processing clumped cells, which leads to the adhesion of cell instances after semantic segmentation, thereby degrading the segmentation accuracy. Furthermore, the performance of existing methods is highly dependent on the manually selected hyperparameters during the separation procedure. Many methods have deployed a large number of hyperparameters that are fine-tuned on pre-existing datasets, which require retuning on new datasets. Meanwhile, some models^{58,59} use shape priors to constrain model predictions to a set of natural variations, but these also require the cells to fall into a particular shape, thus making model generalization difficult. Instead of following the recent trends in developing more accurate and complex semantic segmentation models, we aim to build a simplified model of high robustness and efficiency, and yet achieving comparable performance in 3D cell segmentation.

We propose 3DCellSeg, a novel domain-specific 3D cell instance segmentation model, with four distinctive novelties. A two-stage pipeline is followed. In the first stage, a light-weight CNN is developed to perform semantic segmentation. In the second stage, cell instance segmentation is conducted using a super voxel-based clustering algorithm (see Fig. 1).

The main novelties of 3DCellSeg are summarized in Table 2.

Results

3DCellSeg pipeline. *Semantic segmentation.* Most existing biomedical image processing models, such as V-Net and VoxResNet, are designed for images containing multiple tissues and structures, such as CT or MRI images. However, the content of the cell membrane images is much simpler. We used this domain-specific knowledge to design our light-weight CNN model, 3DCellSegNet, which has a faster inference speed while achieving performance comparable to other state-of-the-art cell segmentation models.

Instance segmentation. Our TASCAN algorithm works on super voxels (small clusters of voxels). Only one hyperparameter in TASCAN (and for the whole pipeline) needs to be set: namely, the minimum touching area between two cell foreground super voxels. TASCAN helps reduce mis-clustering because the super voxels within the same cell usually have much larger touching areas than the super voxels across two neighboring cells. The

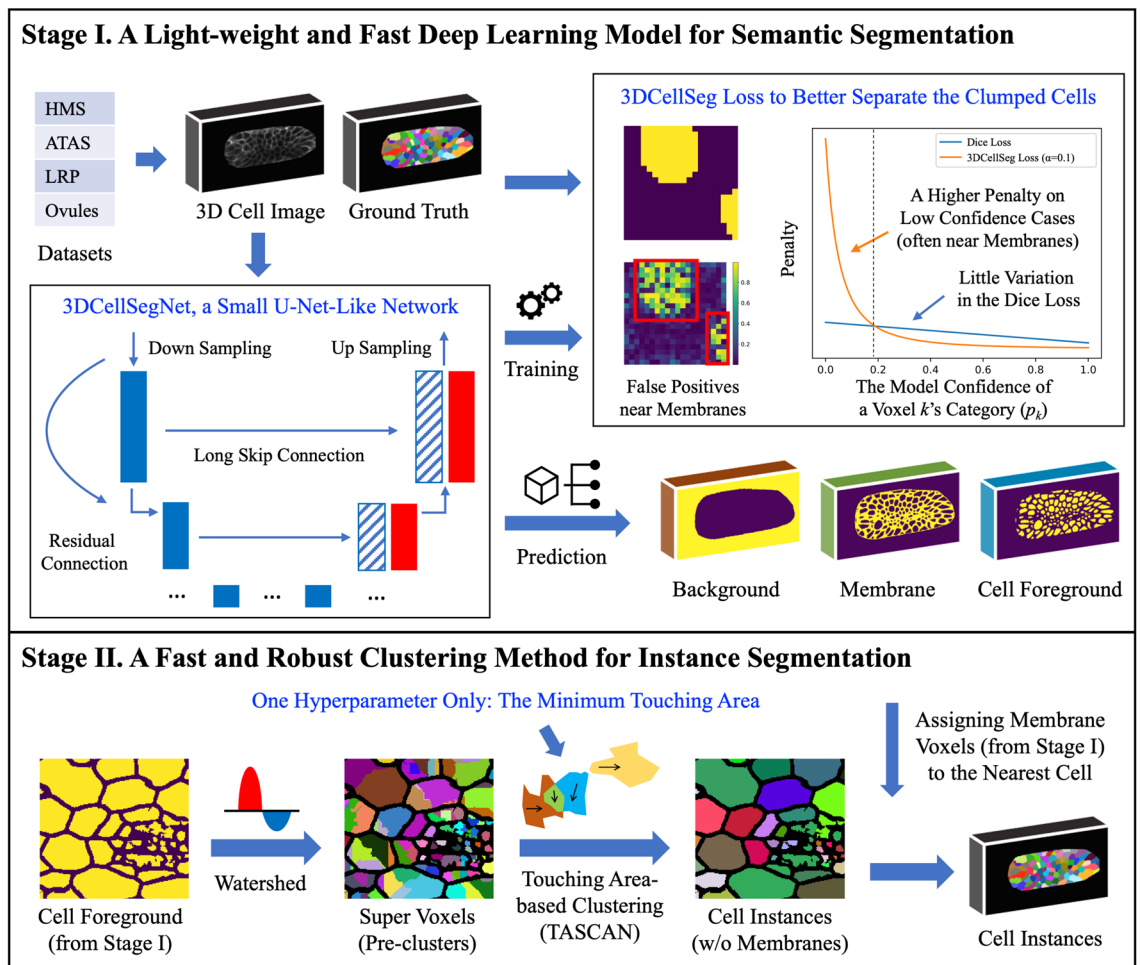


Figure 1. 3DCellSeg: A two-stage light-weight, fast, and robust pipeline for 3D cell segmentation. [Note: There are two stages in the pipeline. The first stage is a semantic segmentation, where the input is a 3D cell membrane image and the output consists of three masks, which indicate whether a voxel is the cell foreground, membrane, or background. The second stage is an instance segmentation performed on the basis of these three masks. The cellular images and segmentation results were generated by Python Matplotlib (<https://matplotlib.org>) using the HMS dataset].

Aspect	Novelty
Network	Based on the characteristics of cell membrane images, a light-weight network, 3DCellSeg, is designed to yield a fast inference speed while achieving an accuracy comparable or superior to the existing cutting-edge approaches
Loss function	A new loss function, 3DCellSeg Loss, is proposed to tackle the clumped cell problem
Post-processing	Inspired by DBSCAN (Density-based Spatial Clustering of Applications with Noise) ⁴⁸ , a new clustering method, TASCAN (Touching Area-based Spatial Clustering of Applications with Noise) is proposed for 3D cell instance segmentation; TASCAN operates faster, achieves better performance, and requires only one single manually selected hyperparameter
Model usability	3DCellSeg pipeline is robust, easy to fine-tune, and outperforms existing cutting-edge methods across different experimental datasets

Table 2. Key novelties of 3DCellSeg.

super voxels across two neighboring cells are caused by membrane voxels being misclassified as the cell foreground, and this misclassified area is usually small in practice.

Loss function. Cells are densely packed across membrane images. Under existing methods, this results in the adhesion of cell masks after semantic segmentation, which greatly degrades the accuracy of instance segmentation. We therefore designed a new loss function, 3DCellSeg Loss, to address the problem. As the top right box in Fig. 1 shows, based on the Dice Loss function, 3DCellSeg adds weight matrices to penalize the voxels that are closer to the cell membranes and replaces p_k (model confidence of a voxel being the cell foreground) with

$\frac{p_k}{p_k + \alpha}$ (see Method for further details). This suppresses the segmentation of the cell foreground when the model confidence is low (which often happens near the cell membranes) and thus reduces adhesions of cell masks.

Four cell membrane image datasets consisting of both animal and plant tissues were used for training and testing: HMS (zebrafish cells); ATAS (Arabidopsis thaliana apical stem cells)⁴⁹; LRP (Arabidopsis thaliana lateral root cells)⁵⁰; and Ovules (Arabidopsis thaliana Ovules cells)⁵¹. Four types of metrics were used for evaluation: Jaccard Index (JI), Dice Similarity Coefficient (DSC), Adapted Rand Error (ARE)⁵², and Variation of Information (VOI)⁵³. A detailed description of the experimental datasets and performance evaluation metrics can be found under Method.

Performance comparison. We compared our 3DCellSeg with both traditional and deep learning methods. To benchmark 3D image performance, we compared 3DCellSeg with 3D U-Net based methods¹¹. To test the model performance on the animal cell membrane images, we retrained U-Net + SWS²¹, U-Net + GASP^{23,40}, U-Net + MultiCut^{23,48}, and U-Net + MutexWS^{23,41} on the HMS dataset, and compared their performance with our method and ACME (traditional method, developed using the HMS dataset) (see Table 2 and Fig. 2). To ensure that our model comparison using the same set of plant cell membrane images is being fairly conducted, we compared our model with U-Net + SWS (built using ATAS) on the ATAS dataset, while U-Net + MutexWS, U-Net + MultiCut, and U-Net + GASP (built using LRP and Ovules) were compared with our method on the LRP and Ovules datasets (see Table 2 and Fig. 3). We used the default hyperparameters of other models (most of the hyperparameters relate to instance segmentation), and used the same hyperparameter (the minimum touching area between two cell foreground super voxels) for our model across the four datasets. We evaluated 3DCellSeg and other baseline models with nine metrics: ARE, VOI_{split}, VOI_{merge}, Avg JI, Avg DSC, JI > 70%, DSC > 70%, JI > 50%, and DSC > 50% (for their definitions, see Datasets and metrics in Method). The results are shown in Table 3. For ARE, VOI_{split}, and VOI_{merge}, the lower the value, the higher the accuracy. For Avg JI, Avg DSC, JI > 70%, DSC > 70%, JI > 50%, and DSC > 50%, a higher value indicates a higher accuracy.

To assess 2D image performance, we ran three object-detection-based instance segmentation models developed for 2D images: two Mask R-CNN²⁸ models with different backbones (ResNet-50-C4 and ResNet-50-FPN⁵⁵, denoted as Mask R-CNN C4 and Mask R-CNN FPN respectively) and a currently-published panoptic segmentation method named PanopticFPN⁵⁶. These models were fed 2D slices of cell membrane images from HMS, ATAS, LRP, and Ovules for model training and testing. ARE, VOI_{split}, VOI_{merge}, JI-related, and DSC-related metrics were calculated for each 2D segmentation method. The results are shown in Table 3. For all the results shown in this article (except for the ablation study for transfer learning in Table 4), the models were re-trained on the same dataset they were tested on. We observe that since segmenting 2D images is intrinsically easier than segmenting 3D images, if the accuracy values of the three object-detection-based models for 2D images are lower than those of 3DCellSeg and other baseline models for 3D images, then the performance deficit of the three object-detection-based models when adapted to segment 3D images will be even larger.

As shown in Table 3, 3DCellSeg achieves the best performance over three datasets, the HMS, ATAS, and LRP, while attaining a comparable performance on the Ovules dataset. U-Net + SWS, U-Net + GASP, U-Net + MultiCut, and U-Net + MutexWS were built and fine-tuned on Arabidopsis thaliana cell membrane images (the ATAS, LRP, and Ovules datasets). However, their performance drops significantly when applied to the zebra fish cell membrane images (the HMS dataset). It has been noted that U-Net + GASP and U-Net + MutexWS perform less satisfactorily as compared to the traditional ACME method on the HMS dataset. This might be related to their reliance on a significant number of manually selected hyperparameters, which are fine-tuned on the membrane images of a certain type of cells, and may not be transferrable to a new type of cells. 3DCellSeg is more robust, as it requires only one hyperparameter.

Moreover, though PanopticFCN, Mask R-CNN FPN, and Mask R-CNN C4 can achieve a high accuracy on the LRP dataset when only a few cell instances are shown on one 2D slice, they perform poorly on the HMS, ATAS, and Ovules datasets when many more cell instances (100–300) are shown on each slice. The results show that 3DCellSeg is more robust while object detection-based instance segmentation method may work much less satisfactorily when the number of target instances to be segmented is large.

Additionally, the size of the 3DCellSegNet (~ 5 MB) adopted in our pipeline is much smaller than the U-Net (~ 15 MB) in U-Net + SWS, and the U-Net (~ 50 MB) in U-Net + GASP, U-Net + MultiCut, and U-Net + MutexWS, with significant reductions in inference time (the inference time of 3DCellSeg is around 50% to 70% of that of other baseline models on the ATAS, LRP, and Ovules datasets).

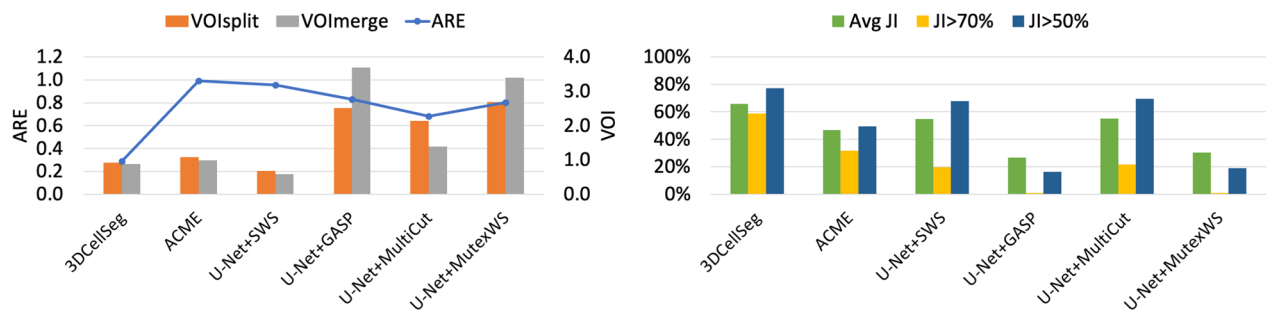
Ablation studies

Ablation studies show that instance segmentation accuracy is improved by our CNN model 3DCellSegNet, our loss function 3DCellSeg Loss, and our clustering algorithm TASCAN.

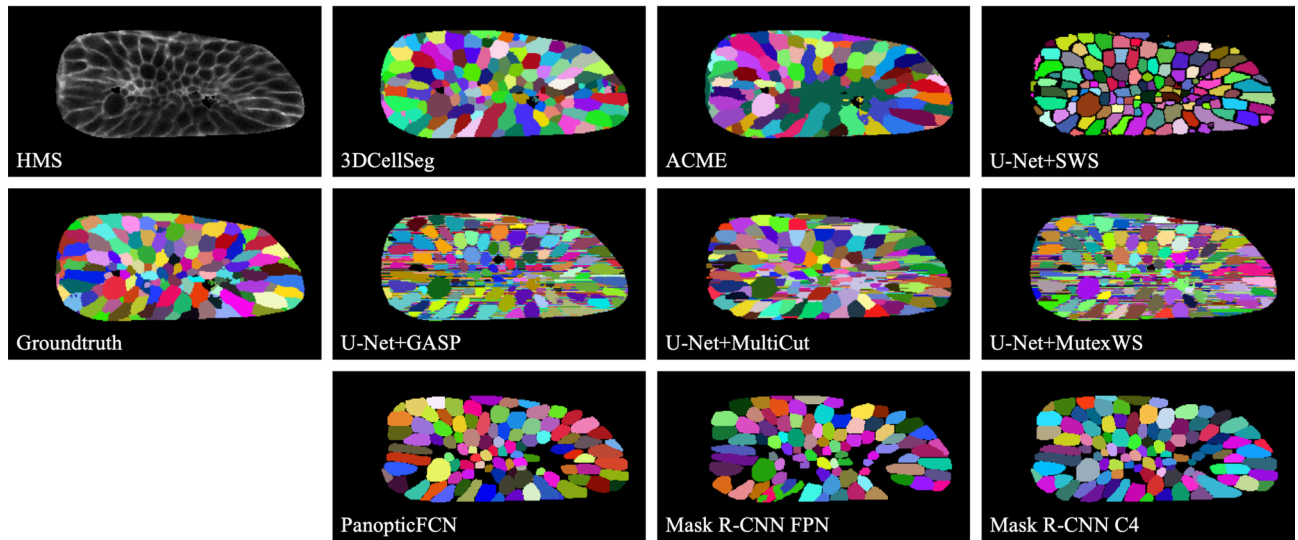
Effects of 3DCellSeg loss. First, we evaluate the effect of the loss function. We compare our 3DCellSeg Loss with the original Dice Loss function (denoted as Dice Loss), Dice Loss with weight matrices (denoted as Dice Loss w/ weights), and 3DCellSeg Loss with the replacement $\frac{p_k}{p_k + \alpha}$ but without weight matrices (denoted as 3DCellSeg Loss w/o weights) on the four datasets (shown in the green-colored rows in Table 4). We find that weight matrices and replacement both greatly improve segmentation accuracy on all metrics. When both are incorporated into the model, the accuracy is further improved.

Effects of 3DCellSegNet. Second, we compare the performance of 3DCellSegNet with that of the two commonly used models for medical image processing, VoxResNet and 3D U-Net, as the backbone CNN in our pipeline (shown in the blue-colored rows in Table 4). We find that all three models have a similar accuracy on the

(a) Performance Comparison (HMS Dataset)



(b) Representative Slices Segmented by 3DCellSeg and Baseline Models (HMS Dataset)



(c) Representative Slices Segmented by 3DCellSeg and Object-detection-based Baseline Models (ATAS Dataset)

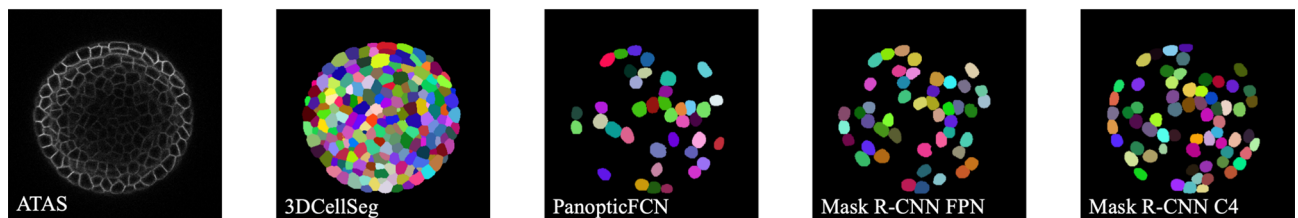


Figure 2. Model comparison and representative slices. [Note (a) shows the accuracies of different cell segmentation models for the HMS dataset. 3DCellSeg achieves the second best accuracy in ARE, VOI_{split} , and VOI_{merge} , and achieves the best accuracy in Avg JI, $JI > 70\%$, and $JI > 50\%$ (the plots for DSC-related metrics are of high similarity to JI-related metrics). (b) and (c) show representative slices of different model segments. ACME tends to under-segment (see the dark green region which mis-classifies different cells as one cell) while U-Net + SWS tends to over-segment (see the over-segmented small cells in the central region). PanopticFCN, Mask R-CNN FPN, and Mask R-CNN C4 are accurate on the HMS dataset but they are severely under-segment on the ATAS dataset. The cellular images in (b) and (c) were generated by Python Matplotlib (<https://matplotlib.org>) using the HMS and ATAS⁴⁹ datasets].

four datasets. However, 3DCellSegNet performs slightly better than the other two models on most metrics. With regards to model size and inference time, 3DCellSegNet (~5 MB) is much smaller than VoxResNet (~30 MB) and 3D U-Net (~50 MB). 3DCellSegNet takes less than 20 s to segment one membrane image from the HMS dataset while the other two require more than one minute.

Effects of TASCAN. Third, we compare the accuracy of adopting TASCAN versus DBSCAN in instance segmentation in our pipeline on the four datasets. This is shown in the orange-colored rows in Table 4. TASCAN performs slightly better than DBSCAN. However, the execution time of TASCAN is less than 20 s while that of DBSCAN is about 180 s. The efficiency advantage of TASCAN becomes more apparent when tackling larger images because the execution time increases cubically with image size. Segmenting larger images like LRP and Ovules takes just a few minutes on TASCAN but more than one hour for a single image on DBSCAN.

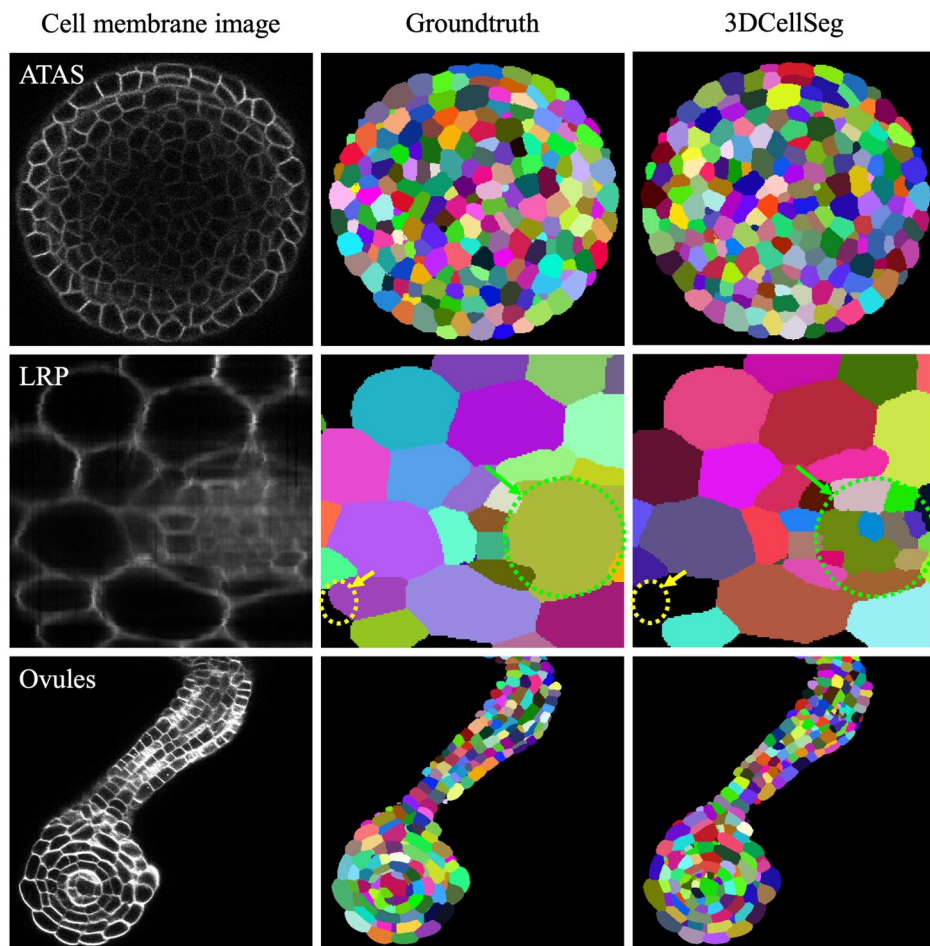


Figure 3. 3DCellSeg performance on the ATAS, LRP, and Ovules datasets. [Note: Different cell instances were randomly assigned different colors. The LRP dataset images are annotated: the yellow circle shows where 3DCellSeg has made a mistake and the green circle shows that 3DCellSeg can segment cells that were not labelled in the ground truth. The cellular images were generated by Python Matplotlib (<https://matplotlib.org>) using the ATAS⁴⁹, LRP⁵⁰, and Ovules⁵¹ datasets].

Transfer learning. The model's transfer learning capability was tested. Based on 3DCellSegNet trained on ATAS dataset, we re-trained the models over only 2 images randomly-picked from the HMS training set. The last row in Table 4 showed that with transfer learning our pipeline still performed well over the HMS dataset, implying that our pipeline was robust and easily applicable to the new cell membrane datasets.

Discussion

Most existing deep learning approaches to 3D cell segmentation follow a two-stage pipeline and focus on either CNN model architecture design in the semantic segmentation stage or post-processing design in the instance segmentation stage. However, the clumped cell problem in cell segmentation has not been adequately addressed, with challenges in accuracy and robustness as existing segmentation pipelines rely on extensive manually selected hyperparameters. To tackle these critical but overlooked problems, we propose a deep learning-based two-stage pipeline, 3DCellSeg. The whole pipeline requires only one manually selected hyperparameter (the minimum touching area between two super voxels of the cell foreground). In the first stage, a light-weight CNN-based U-Net-like model, 3DCellSegNet, performs semantic segmentation while addressing the clumped cell problem by incorporating a novel loss function, 3DCellSeg Loss, into the network training process. In the second stage, a novel touching area-based clustering algorithm, TASCAN, distinguishes cell instances from the other types of semantic voxels according to the minimum touching area.

The experimental results based on four animal and plant cell membrane image datasets show the performance of the 3DCellSeg pipeline to be better than or at least comparable to other approaches. For the HMS (animal), ATAS (plant), and LRP (plant) datasets, 3DCellSeg pipeline achieved better performance than the traditional baseline (ACME) and the U-Net-based deep learning baselines in terms of both overall accuracy and cell count accuracy. For the Ovules (plant) dataset, 3DCellSeg pipeline achieved performance comparable to the U-Net-based deep learning baselines that were initially built and fine-tuned based on the datasets of *Arabidopsis thaliana*

Dataset	Method	ARE	VOI _{split}	VOI _{merge}	Avg JI	Avg DSC	JI>70%	DSC>70%	JI>50%	DSC>50%
HMS	3DCellSeg	0.29	0.92	0.89	65.7%	76.4%	58.9%	74.2%	77.1%	88.8%
	ACME	0.99	1.08	0.99	46.9%	57.9%	31.8%	47.3%	49.3%	65.4%
	U-Net+SWS	0.96	0.68	0.58	54.8%	68.7%	19.5%	62.2%	68.0%	86.5%
	U-Net+GASP	0.83	2.52	3.69	26.7%	39.1%	1.0%	12.5%	16.2%	30.8%
	U-Net+MultiCut	0.68	2.13	1.39	55.2%	68.8%	21.6%	63.4%	69.6%	86.4%
	U-Net+MutexWS	0.80	2.69	3.40	30.2%	43.6%	1.0%	14.3%	19.1%	38.6%
	PanopticFCN	0.60	0.71	0.58	50.7%	62.3%	35.1%	50.0%	52.6%	64.5%
	Mask R-CNN FPN	0.59	0.72	0.58	50.9%	60.5%	39.3%	54.2%	57.7%	70.0%
Mask R-CNN C4	0.52	0.80	0.56	54.3%	64.6%	40.8%	58.0%	61.8%	75.9%	
ATAS	3DCellSeg	0.16	0.39	0.51	92.3%	95.6%	96.6%	97.4%	97.8%	98.9%
	U-Net+SWS	0.48	0.40	0.47	87.0%	93.1%	81.5%	87.4%	88.7%	95.8%
	ACME	0.73	0.64	0.86	80.2%	89.0%	65.8%	70.6%	81.4%	90.7%
	PanopticFCN	0.91	0.58	0.79	24.8%	34.7%	4.2%	10.4%	12.7%	34.9%
	Mask R-CNN FPN	0.74	0.39	0.62	37.2%	42.6%	32.8%	40.2%	41.3%	45.1%
	Mask R-CNN C4	0.73	0.45	0.58	39.9%	46.8%	32.1%	42.9%	44.8%	51.2%
LRP	3DCellSeg	0.08	0.34	0.40	67.9%	74.7%	65.1%	71.1%	73.1%	78.8%
	U-Net+GASP	0.07	0.62	0.41	64.4%	74.0%	52.2%	64.9%	70.5%	83.6%
	U-Net+MultiCut	0.09	0.34	0.53	66.2%	74.4%	59.1%	67.4%	70.2%	83.1%
	U-Net+MutexWS	0.08	0.34	0.53	65.0%	73.2%	57.5%	67.7%	70.3%	79.9%
	PanopticFCN	0.27	0.46	0.24	59.3%	68.0%	52.2%	58.8%	60.9%	69.5%
	Mask R-CNN FPN	0.11	0.26	0.17	78.5%	83.8%	78.4%	86.0%	88.0%	90.4%
	Mask R-CNN C4	0.16	0.31	0.19	76.2%	82.2%	76.2%	84.5%	86.8%	89.7%
Ovules	3DCellSeg	0.25	0.53	0.59	74.7%	82.2%	77.4%	82.7%	84.0%	89.6%
	U-Net+GASP	0.16	0.63	0.46	77.5%	82.9%	79.4%	85.3%	86.0%	88.9%
	U-Net+MultiCut	0.15	0.54	0.46	78.8%	83.7%	81.9%	86.0%	86.5%	88.9%
	U-Net+MutexWS	0.16	0.58	0.48	77.4%	82.7%	79.5%	84.9%	85.6%	88.4%
	PanopticFCN	0.61	0.51	0.42	46.6%	56.1%	36.5%	48.6%	50.6%	57.9%
	Mask R-CNN FPN	0.59	0.48	0.37	48.1%	55.0%	42.4%	51.9%	54.0%	61.2%
	Mask R-CNN C4	0.52	0.58	0.39	51.5%	59.5%	44.1%	55.5%	58.0%	66.9%

Notes

3D segmentation models

2D segmentation models

Bold Denotes the best performance amongst all tested models for each dataset

Table 3. Comparison of model performance on the HMS, ATAS, LRP, and Ovules datasets. [Note: For the HMS dataset, U-Net + SWS, U-Net + GASP, U-Net + MultiCut, and U-Net + MutexWS were retrained on default hyperparameters, and compared with our 3DCellSeg. For the ATAS dataset, U-Net + SWS, which was originally developed, trained and fine-tuned on the ATAS dataset, was compared with 3DCellSeg. For the LRP and Ovules datasets, U-Net + GASP, U-Net + MultiCut, and U-Net + MutexWS, which were originally built, trained and fine-tuned on LRP and Ovules, were compared with 3DCellSeg. Object-detection based instance segmentation methods (PanopticFCN, Mask R-CNN FPN, and Mask R-CNN C4) trained on 2D slices of the HMS, ATAS, LRP, and Ovules datasets were also taken as baselines for model comparison. ARE, VOI_{split}, VOI_{merge}, JI-related, and DSC-related metrics were calculated on 3D space for 3DCellSeg, ACME, U-Net + SWS, U-Net + GASP, U-Net + MultiCut, and U-Net + MutexWS and were calculated on 2D slices for PanopticFCN, Mask R-CNN FPN, and Mask R-CNN C4].

Dataset	Ablation	ARE	VOI _{split}	VOI _{merge}	Avg JI	Avg DSC	JI>70%	DSC>70%	JI>50%	DSC>50%
HMS	3DCellSeg Loss	0.29	0.92	0.89	65.7%	76.4%	58.9%	74.2%	77.1%	88.8%
	Dice loss	0.54	1.72	1.68	36.7%	44.8%	28.9%	33.2%	34.7%	45.2%
	Dice Loss w/ weights	0.31	1.03	0.96	61.2%	72.1%	51.5%	64.4%	67.2%	82.7%
	3DCellSeg Loss w/o weights	0.33	1.05	1.01	59.2%	70.4%	48.6%	60.6%	63.6%	80.5%
	3DCellSegNet	0.29	0.92	0.89	65.7%	76.4%	58.9%	74.2%	77.1%	88.8%
	VoxResNet	0.33	1.12	0.97	62.8%	74.0%	54.0%	71.6%	73.7%	85.8%
	3D U-Net	0.34	1.14	0.98	64.6%	76.3%	54.2%	74.4%	77.2%	90.2%
	TASCAN	0.29	0.92	0.89	65.7%	76.4%	58.9%	74.2%	77.1%	88.8%
DBSCAN	0.29	0.94	0.91	65.2%	75.7%	60.8%	72.5%	74.6%	86.9%	
ATAS	3DCellSeg Loss	0.16	0.39	0.51	92.3%	95.6%	96.6%	97.4%	97.8%	98.9%
	Dice loss	0.46	0.41	0.84	79.8%	85.9%	81.1%	83.3%	83.5%	87.5%
	Dice Loss w/ weights	0.22	0.40	0.51	87.3%	91.7%	90.9%	92.8%	93.4%	94.2%
	3DCellSeg Loss w/o weights	0.23	0.41	0.54	86.8%	90.5%	89.8%	92.4%	93.5%	94.6%
	3DCellSegNet	0.16	0.39	0.51	92.3%	95.6%	96.6%	97.4%	97.8%	98.9%
	VoxResNet	0.16	0.41	0.52	91.1%	94.8%	95.4%	96.5%	97.1%	98.7%
	3D U-Net	0.16	0.40	0.51	91.8%	95.1%	95.9%	97.1%	97.2%	98.8%
	TASCAN	0.16	0.39	0.51	92.3%	95.6%	96.6%	97.4%	97.8%	98.9%
DBSCAN	0.17	0.42	0.54	90.5%	93.2%	94.7%	95.9%	96.9%	97.2%	
LRP	3DCellSeg Loss	0.08	0.34	0.40	67.9%	74.7%	65.1%	71.1%	73.1%	78.8%
	Dice loss	0.25	0.77	0.32	59.3%	64.6%	58.8%	62.2%	63.4%	67.2%
	Dice Loss w/ weights	0.14	0.49	0.48	64.2%	70.6%	60.6%	66.6%	69.0%	75.3%
	3DCellSeg Loss w/o weights	0.24	0.56	0.21	60.5%	67.6%	57.3%	61.2%	67.8%	74.5%
	3DCellSegNet	0.08	0.34	0.40	67.9%	74.7%	65.1%	71.1%	73.1%	78.8%
	VoxResNet	0.11	0.45	0.49	62.4%	69.0%	59.3%	65.4%	66.6%	72.1%
	3D U-Net	0.13	0.47	0.41	64.6%	70.3%	63.3%	69.2%	69.8%	74.4%
	TASCAN	0.08	0.34	0.40	67.9%	74.7%	65.1%	71.1%	73.1%	78.8%
DBSCAN	0.10	0.40	0.45	65.3%	72.9%	64.0%	69.1%	71.7%	76.1%	
Ovules	3DCellSeg Loss	0.25	0.53	0.59	74.7%	82.2%	77.4%	82.7%	84.0%	89.6%
	Dice loss	0.46	0.68	0.68	65.7%	73.7%	63.1%	69.9%	71.5%	80.1%
	Dice Loss w/ weights	0.45	0.51	0.72	70.3%	77.8%	68.9%	75.1%	76.6%	84.0%
	3DCellSeg Loss w/o weights	0.34	0.53	0.65	71.5%	79.2%	70.2%	76.1%	77.8%	86.0%
	3DCellSegNet	0.25	0.53	0.59	74.7%	82.2%	77.4%	82.7%	84.0%	89.6%
	VoxResNet	0.39	0.54	0.69	70.3%	77.7%	69.4%	75.3%	76.8%	83.1%
	3D U-Net	0.23	0.51	0.66	73.4%	80.6%	73.6%	79.4%	80.7%	86.8%
	TASCAN	0.25	0.53	0.59	74.7%	82.2%	77.4%	82.7%	84.0%	89.6%
DBSCAN	0.26	0.57	0.63	72.1%	81.0%	75.7%	81.6%	82.1%	87.2%	
Transfer Learning		0.35	1.10	1.07	55.1%	66.9%	38.4%	58.9%	61.8%	77.1%
Notes										
		Ablation study on the loss function.								
		Ablation study on the network.								
		Comparison of TASCAN and DBSCAN.								

Table 4. Ablation studies showing 3DCellSeg Loss, 3DCellSegNet, TASCAN, and Transfer Learning.

cells, including the Ovules dataset. However, some of these U-Net-based deep learning baselines performed even worse than the ACME method on the HMS dataset (zebrafish cells), likely due to the use of extensive

hyperparameters that are not transferrable across different cell types. In contrast, even without fine-tuning, 3DCellSeg still performed satisfactorily across the four different datasets because it has a simple structure and only utilizes one hyperparameter.

Moreover, 3DCellSeg is more robust than the object-detection-based deep learning baselines when the number of cell instances to be segmented is large. As compared to 3DCellSeg, when the number of cell instances was small on each slice (tens of cells), the object-detection-based deep learning baselines (including Mask R-CNN FPN and Mask R-CNN C4) achieved a higher accuracy on the LRP dataset. However, when the number of cell instances became large on each slice (around 100–300 cells), these object-detection-based baselines failed to accurately segment cells from the HMS and ATAS datasets.

Our ablation study reveals how the four components of the 3DCellSeg pipeline perform relative to their counterparts across four different datasets. Firstly, the use of 3DCellSeg Loss improved the segmentation accuracy of the original Dice Loss function for the four datasets, especially for the LRP and Ovules datasets with a higher cell shape irregularity. This is because 3DCellSeg Loss was designed specifically for cell segmentation and can suppress the cell foreground segmentation near the cell membranes where model confidence is not sufficiently high, thus reducing misclassification of voxels and enabling more accurate cell instance clustering.

Secondly, 3DCellSegNet slightly outperformed two CNN backbones commonly used in CT or MRI image segmentation (VoxResNet and 3D U-Net) for most performance metrics across the four datasets. This result suggests that a simple CNN structure with fewer parameters can fully capture the characteristics of cell membrane images that have a more recurrent structure than CT and MRI images. The simplified CNN structure can reduce the risk of model overfitting when limited training samples are available, thus providing a more robust backbone for cell segmentation across different datasets. In addition to improved accuracy, the light-weight design of 3DCellSegNet enables it to deliver much faster training and segmentation than the CNN models implemented in other pipelines. With 3DCellSegNet, a HMS image of 5 MB takes < 20 s to process, as compared to one of 30 MB or larger, which takes ~ 60 s to process, using VoxResNet or 3D U-Net.

Thirdly, the touching area-based clustering method, TASCAN, is slightly more accurate than the original clustering method DBSCAN. The marginal improvement in accuracy is probably due to the fact that most misclassified foreground masks have already been addressed by the 3DCellSeg Loss function in the first stage. However, TASCAN has a higher robustness and is easier to implement as it requires only one hyperparameter. In addition, the computational complexity of TASCAN is much lower because it processes super voxels rather than voxels. When dealing with larger images such as those in the LRP and Ovules datasets, TASCAN can significantly reduce the clustering time from one hour to a few minutes in the second stage.

Fourthly, the transfer learning experiment demonstrates the robustness of the whole 3DCellSeg pipeline across different datasets. When re-purposing a trained 3DCellSegNet model to segment cells in different cell membrane images, our 3DCellSeg pipeline still performs well, despite the fact that only a few new images are available for re-training.

Overall, as compared to other existing approaches to 3D cell segmentation, our approach is superior in three aspects: Firstly, our approach champions the use of domain-specific modelling to better capture the characteristics of the 3D cell images. We have utilized a light-weight CNN model with fewer parameters to perform semantic segmentation based on the knowledge that cell membrane images contain more recurrent patterns than CT or MRI images. Meanwhile, we have designed a custom loss function to address the misclassification challenge due to clumped cells. Secondly, our approach is robust and easy to fine-tune, requiring only one hyperparameter. Minimizing the number of hyperparameters is especially beneficial when adapting the pipeline to a new dataset, as the setting of hyperparameters is often not transferable across different datasets. Thirdly, the experimental results have demonstrated that our proposed method achieves performance better than or similar to state-of-the-art models but is more computationally efficient. They also suggest that our novel 3DCellSeg pipeline can accurately process high-throughput imaging data, allowing the automatic detection of 3D cells in a large scale in real-time, thus paving the way for cellular disease mechanism discovery.

Our work can be improved in two aspects. First, given that the CNN model is only used in semantic segmentation, we are yet to fully exploit deep learning models in learning representations of high-dimensional 3D images. In the future, we will build an end-to-end deep learning model to output the 3D cell instance segmentation directly. Second, despite the fact that we have accounted for the clumped cell issue, more domain-specific knowledge of the cell characteristics, such as the size and distribution of cells, can be integrated into our pipeline to improve cell identification/classification accuracy.

Method

3DCellSeg performs semantic segmentation to identify the cell foreground masks, followed by instance segmentation. Two challenges of cell segmentation are addressed. The first is the clumped cell problem. Adhesions of the cell foreground masks need to be addressed during semantic segmentation, as they reduce the accuracy of instance segmentation. The second is the selection of hyperparameters, which is currently performed manually on a trial-and-error basis across a particular dataset, and may not be transferable across other datasets.

Experimental setup. Consider a training set \mathcal{D} of pairs $\{(x_i, s_i)\}_{i=1}^N$ where x_i is a 3D cell membrane image and s_i is the corresponding segmentation, where different integers represent different cell instances. Furthermore, consider a segmentation pipeline $h(f)_{\theta, \lambda}$, where f denotes semantic segmentation, h denotes instance segmentation, θ denotes the CNN parameters to be trained, and λ denotes the manually set hyperparameter(s).

Dataset and metric. Four datasets containing 3D cell membrane images and corresponding voxel-wise cell labels were used for model training and testing. The first dataset, labelled as HMS, contains images of zebrafish

cells. It is a new open-source dataset compiled by the Department of Systems Biology of Harvard Medical School. There are 36 images with a size of $181 \times 331 \times 160$. A detailed description of HMS dataset is shown in the last subsection of Method. The second dataset, labelled as ATAS⁴⁹, is an open-source dataset containing membrane images of Arabidopsis thaliana apical stem cells. It contains 126 cell membrane images with a size of $224 \times 512 \times 512$. The third dataset, labelled as LRP⁵⁰, is generated from three time-lapse videos documenting how Arabidopsis thaliana lateral root primordia developed. 27 images of the size of $2000 \times 1000 \times 500$ are labelled voxel by voxel. The fourth dataset, labelled as Ovules⁵¹, contains 48 images of size $500 \times 1000 \times 1000$ documenting all development stages of Arabidopsis thaliana ovules.

Our CNN model was trained for semantic segmentation, to generate the cell foreground, membrane, and background masks \mathbf{g}_i^c (every voxel in \mathbf{g}_i^c is 0 or 1; c represents a category) for each image \mathbf{x}_i . Additionally, to address the clumped cell problem, in the loss function we use a weight matrix \mathbf{w}_i^c that assigns a larger weight to voxels closer to the cell membranes. \mathbf{w}_i^c is calculated by applying a reverse distance transform⁵⁷ to \mathbf{g}_i^c . To reduce memory usage, we cropped \mathbf{x}_i , \mathbf{g}_i^c , and \mathbf{w}_i^c to small cuboids $\{\mathbf{x}_i^j\}$, $\{\mathbf{g}_i^{c,j}\}$, and $\{\mathbf{w}_i^{c,j}\}$, and trained the CNN model on cuboids $\left\{ \left\{ \left(\mathbf{x}_i^j, \mathbf{g}_i^{c,j}, \mathbf{w}_i^{c,j} \right) \right\}_{j=1}^M \right\}_{i=1}^N$.

We evaluated the performance of the segmentation based on four types of metrics: Jaccard Index (JI), Dice Similarity Coefficient (DSC), Adapted Rand Error (ARE), and Variation of Information (VOI). JI and DSC values were calculated for each cell, measuring the ratio of the correctly predicted voxels. We evaluated performance based on overall accuracy and cell count accuracy. Overall accuracy refers to the average values of JI and DSC of all cells (denoted as Avg JI and Avg DSC, respectively). Cell count accuracy is the fraction of cells whose JI (or DSC) is more than 0.7 (or 0.5). ARE measures how much the algorithm outperforms a random model⁵², and VOI is an entropy-based measure of clustering quality⁵³. $\text{VOI}_{\text{split}}$ measures split errors and $\text{VOI}_{\text{merge}}$ measures merge errors.

$$JI(seg, gt) = \frac{|seg \cap gt|}{|seg \cup gt|} \quad (1)$$

$$DSC(seg, gt) = \frac{2|seg \cap gt|}{|seg| + |gt|} \quad (2)$$

$$ARE = 1 - \text{maxScore}(\text{Rand Index}) \quad (3)$$

$$VOI(seg, gt) = 2H(seg \cap gt) - H(seg) - H(gt) \quad (4)$$

where seg is the model prediction, gt is the ground truth mask, and H is the conditional entropy function. JI is the ratio of the intersection of seg and gt over the union of them while DSC is the ratio of two times of the intersection over the sum of the size of seg and gt . ARE is derived from the Rand Index and VOI is an entropy-based measure. Please see the complete definitions of ARE and VOI in⁵² and⁵³ respectively.

3DCellSegNet for semantic segmentation. Figure 4 shows the structure of 3DCellSegNet. Based on the knowledge that cell membrane images contain more recurrent patterns than CT or MRI images, we built a lighter weight network for medical image processing which achieves faster inference speed with similar performance. For example, the average inference time of 3DCellSegNet on HMS and LRP is 20 and 50 s respectively, while the inference time of 3D U-Net and VoxResNet is around 2 times and 1.5 times longer. 3DCellSegNet has a shallow U-Net like shape with residual connection at the down sampling stage. Intuitively, for cell membrane images, only local features are required to determine the voxel category. In addition, by removing extra voxels on the edge of the feature maps during up-sampling, 3DCellSegNet can process cuboid-shape images of any size. We trained the model by feeding cuboids of slightly different sizes, thus improving the segmentation of cuboid interface and the model's robustness.

We implemented 3DCellSegNet with PyTorch⁵⁸. The model was trained on one Nvidia GeForce RTX 2080 Ti with a cuboid size ranging from $56 \times 56 \times 56$ to $64 \times 64 \times 64$. For the four datasets, the longest training time is 48 h, using Adam optimizer and a batch size of 7.

3DCellSeg Loss to tackle the clumped cell problem. Since cells are densely packed in the HMS dataset, there is an adhesion of the cell foreground masks in the semantic segmentation, thereby decreasing the accuracy of the instance segmentation. We proposed our 3DCellSeg Loss L to tackle the problem.

$$L_{\text{Others}} = 1 - \frac{2 \sum_{k=1}^K p_k g_k w_k}{\sum_{k=1}^K p_k^2 + \sum_{k=1}^K g_k^2} \quad (5)$$

$$L_{\text{CellForeground}} = 1 - \frac{2 \sum_{k=1}^K \left(\frac{p_k}{p_k + \alpha} \right) g_k w_k}{\sum_{k=1}^K \left(\frac{p_k}{p_k + \alpha} \right)^2 + \sum_{k=1}^K g_k^2} \quad (6)$$

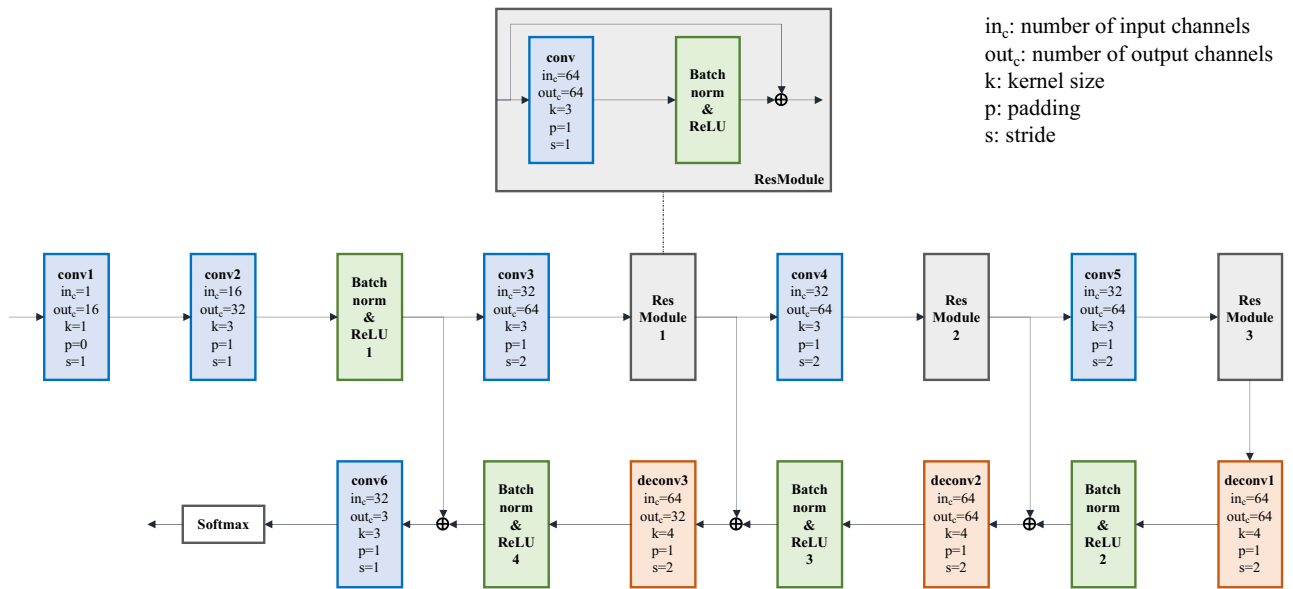


Figure 4. The structure of 3DCellSegNet. [Note: The extra voxels on the edge of the feature maps are removed after each deconvolution operation, in order to ensure the size of the up-sampled feature map is identical with that of the corresponding down-sampled feature map].

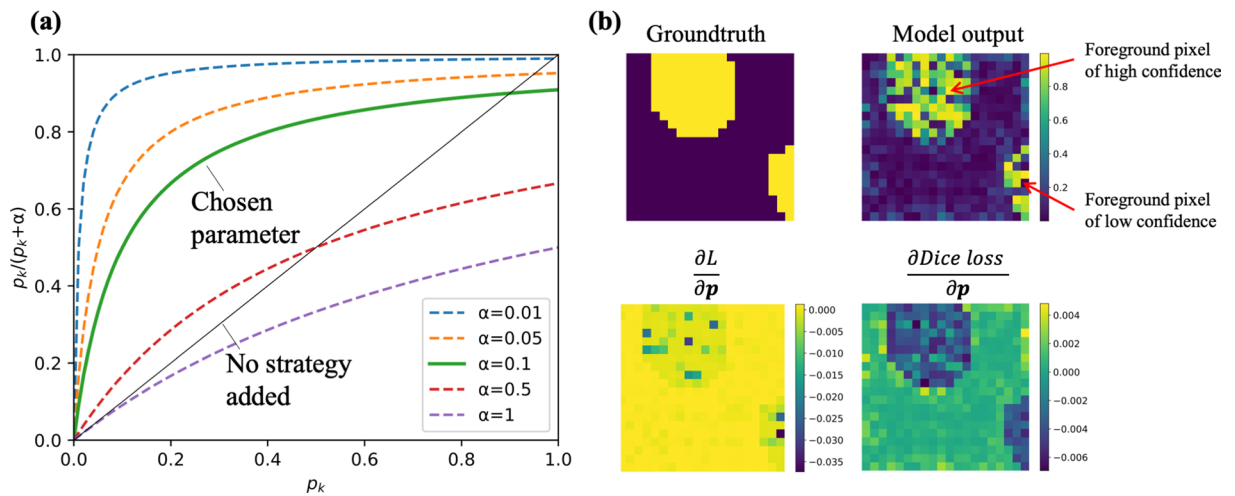


Figure 5. Addressing the clumped cell problem using 3DCellSeg Loss. [Note (a) $\frac{p_k}{p_k+\alpha}$ vs p_k at different α values shows how different α affects the replacement $\frac{p_k}{p_k+\alpha}$. (b) A 2D slice of a simulation illustrating the difference between Dice Loss $\frac{\partial L}{\partial p}$ and 3DCellSeg Loss $\frac{\partial \text{DiceLoss}}{\partial p}$. The simulation results in (b) were generated by Python Matplotlib (<https://matplotlib.org>)].

$$L = L_{\text{CellForeground}} + L_{\text{Others}} \tag{7}$$

where $p_k \in [0, 1]$ is the model confidence of a voxel being the cell foreground; $g_k \in \{0, 1\}$ is the voxel value of the category mask; w_k is the weight of a voxel k (w_k is larger if k is closer to cell membrane); K is the number of voxels; α is a constant; $L_{\text{CellForeground}}$ is the loss function for the cell foreground and L_{Others} is the loss function for other categories.

We used weight matrix w and replacement $\frac{p_k}{p_k+\alpha}$ to suppress the segmentation of the cell foreground close to the cell membrane if the confidence is not high, as the false positives of foreground voxels will result in clumped cell instances. 3DCellSeg Loss works like a regularizer, penalizing low confidence. Figure 5a shows the values of $\frac{p_k}{p_k+\alpha}$ at different α values. We chose 0.1 to achieve our desired regularization effect: when p_k is small, the increasing rate of $\frac{p_k}{p_k+\alpha}$ should be high; when p_k is near 0.5, $\frac{p_k}{p_k+\alpha}$ grows slowly as p_k increases; however, even when p_k is near 1, $\frac{p_k}{p_k+\alpha}$ should be slightly less than 1, thus maintaining the penalty.

$\frac{\partial L}{\partial p}$ indicates that 3DCellSeg Loss penalizes heavily the positive voxels of low confidence (< 0.2), but penalizes slightly other positive voxels. $\frac{\partial \text{DiceLoss}}{\partial p}$ penalizes the voxels on an equal basis. Hence, 3DCellSeg Loss will not push positive voxels of medium confidence (0.2–0.5) to a more positive end. The derivative of 3DCellSeg Loss $\frac{\partial L}{\partial p_k}$

Algorithm 1 TASCAN for Cell Clustering

Require: A set of unprocessed super-voxels generated by watershed $\mathbb{U} = \{u\}$, the surface area of each super-voxel S_u , a threshold value $minArea = 30$ in our application.

- 1: **for** u in \mathbb{U} **do**
- 2: **repeat**
- 3: $V_u = getNeighbourSuperVoxels(u, \mathbb{U})$
- 4: **for** v in V_u **do**
- 5: Calculate the touching area S_{uv}
- 6: **if** $S_{uv} > minArea$ or $S_{uv}/S_u > 0.5$ or $S_{uv}/S_v > 0.5$ **then**
- 7: Merge v into u
- 8: Remove v from \mathbb{U}
- 9: Remove v from V_u
- 10: **end if**
- 11: **end for**
- 12: **until** no super-voxel merged in current repetition
- 13: Remove u from \mathbb{U}
- 14: **end for**

Figure 6. TASCAN algorithm for cell clustering.

$(\frac{\partial L}{\partial p_k} = \frac{\partial L_{CellForeground}}{\partial p_k} + \frac{\partial L_{Others}}{\partial p_k})$ provides a numerical explanation on how 3DCellSeg Loss works by comparing how $\frac{\partial L}{\partial p_k}$ and $\frac{\partial DiceLoss}{\partial p_k}$ change with p_k in the top right box in Fig. 1.

$$\frac{\partial L_{Others}}{\partial p_k} = 2 \frac{-g_k w_k \left(\sum_{l=1}^K p_l^2 + \sum_{l=1}^K g_l^2 \right) + 2p_k \sum_{l=1}^K p_l g_l w_l}{\left(\sum_{l=1}^K p_l^2 + \sum_{l=1}^K g_l^2 \right)^2} \quad (8)$$

$$\frac{\partial L_{CellForeground}}{\partial p_k} = 2 \frac{-g_k w_k \left(\sum_{l=1}^K \left(\frac{p_l}{p_l + \alpha} \right)^2 + \sum_{l=1}^K g_l^2 \right) + 2 \left(\frac{p_k}{p_k + \alpha} \right) \sum_{l=1}^K \left(\frac{p_l}{p_l + \alpha} \right) g_l w_l}{\left(\sum_{l=1}^K \left(\frac{p_l}{p_l + \alpha} \right)^2 + \sum_{l=1}^K g_l^2 \right)^2} \cdot \frac{\alpha}{(p_k + \alpha)^2} \quad (9)$$

In Fig. 1, the orange line indicates $\frac{\partial L}{\partial p_k}$ while the blue line indicates $\frac{\partial DiceLoss}{\partial p_k}$. The two lines represent a simulation case where the model confidence of a cell foreground voxel p_k varies from 0 to 1. As the figure shows, $\frac{\partial DiceLoss}{\partial p_k}$ follows a linear trend as p_k changes. However, $\frac{\partial L}{\partial p_k}$ imposes a heavy penalty for low confidence cases ($p_k < 0.2$) but only a light penalty for high confidence cases (where the penalty is lower than $\frac{\partial DiceLoss}{\partial p_k}$). This shows that 3DCellSeg Loss works like a regularizer that penalizes model predictions of low confidence. The regularization effect of 3DCellSeg Loss is also illustrated in Fig. 5b which shows a 2D simulation. With the ground truth, where the yellow round shapes represent the cell foreground and the purple region represents the cell background, a simulated model output is generated. The lightness of the color indicates the confidence of a pixel being identified as a cell. Based on the ground truth and model output, the derivatives of Dice Loss and 3DCellSeg Loss over the 2D slice are shown in the images labelled $\frac{\partial L}{\partial p}$ and $\frac{\partial DiceLoss}{\partial p}$ respectively. Significantly, the mis-classified voxels around the edge are penalized more heavily than those in the center. Since mis-classification of the cell foreground often occurs near cell membranes, 3DCellSeg Loss helps address the adhesions of the cell foreground masks.

TASCAN for instance segmentation. After the voxels were assigned different semantic labels, we applied Algorithm 1 to the foreground voxels to divide them into different cell instances. The proposed TASCAN is inspired by DBSCAN⁴⁸. However, our clustering algorithm is much more efficient. First, we applied watershed algorithm to pre-cluster the foreground voxels into super voxels, i.e., small clusters of voxels. Second, TASCAN was used to merge those super voxels that share a large touching area. The details of the algorithm are shown below.

Let u be a super voxel. S_u denotes the area of its surface. V_u denotes the set of all super voxels that touches u . For a super voxel $v \in V_u$, the touching area of u and v is denoted as S_{uv} . We also define a threshold value $minArea$. If the touching area of two super voxels is smaller than $minArea$, they are taken as two separate cells. We set the value of $minArea$ at 30 for the four different datasets. This value is relatively robust and intuitively decided by the size of holes on the membrane (the area of the cell membrane that is mis-classified as the cell interior) in relation to the inference error generated by the backbone neural network. Basically, for an image with a much larger cell size, the value should be increased.

Algorithm 1 (see Fig. 6) shows the operation of TASCAN. The watershed algorithm is used to generate small clusters within the normal cells. Small super voxels are merged with the surrounding super voxels if the touching area is greater than half of its surface area. The physical interpretation of setting the threshold to 0.5 is that if at least half of the surface of one voxel is surrounded by another voxel, these voxels will be merged. After TASCAN clustering has been performed, the unassigned foreground voxels are assigned to their nearest cells. In addition, the membrane voxels are also assigned to their nearest cells. Since the metric for judging whether cell u and v should be merged is symmetrical in terms of u and v , TASCAN does not rely on the order of super voxels; the voxels in each experiment are processed in a random order.

TASCAN operates at a higher speed than DBSCAN. It is capable of processing a large number of voxels all at once, instead of processing each voxel individually. Additionally, TASCAN is good at separating clumped cells. Although our revised loss function can improve the accuracy of recognizing the cell membranes, membrane voxels can still be mis-classified as foreground pixels, since the foreground voxel clusters may be connected with each other. Whereas DBSCAN finds it difficult to separate these clusters if the search radius utilized is either too small or too large (exceeding the thickness of a membrane), TASCAN is able to resolve this problem. The clusters can mostly be separated firstly by the watershed algorithm, followed by separation performed by TASCAN as the connection tunnels are usually very thin.

HMS dataset. Zebrafish transgenic embryos expressing nuclear-localized tomato and membrane-localized citrine (Tg(actb2:Hsa.H2B-tdTomato); Tg(actb2:mem-citrine)hm32,33), Tg(actb2:mem-citrine-citrine)hm30 were used to capture a time-lapse confocal dataset from 12–45 h postfertilization of development. High-resolution imaging was performed with Zeiss 710 confocal microscope with a Plan-Apochromat 40X 1.2 NA objective and a 514 nm laser (20mW, 3% laser power), pixel dwell time: 1.58 μ s; pinhole size: 89 μ m; line averaging: 1; image spacing: 0.2 \times 0.2 μ m, and 1024 \times 1024 pixels per image, with an interval of 1.0 μ m through Z for 80 μ m, and temporal resolution of 2 min. A total of 225 timepoints were collected and 32 z-stacks were selected at regular intervals of 10 min for generating ground truth. The otic vesicle and inner luminal surfaces were manually contoured in 2D using Gofigure2 and ITK-Snap. The surfaces were reconstructed in 3D using Powercrust reconstruction algorithm⁵⁹. Since the otic vesicle is ellipsoidal, the images were rotated in 2D so that principal axes were aligned with coordinate axes. The 3D volumetric datasets were then cropped into a smaller size and resampled to have near isotropic sampling ratios (0.4 \times 0.4 \times 0.5 μ m). To establish the ground truth dataset, cells were segmented using ACME. Semantic segmentation images were evaluated in 3D by overlaying on raw image data in GoFigure2 and stepping through z-stacks. Three types of segmentation errors were corrected. Missing cells, over- and under-segmented cells were identified, re-seeded, and the segmentations were re-generated. The generated segmentations were then evaluated manually a second time to guarantee highly accurate ground truth data. For example, the otic vesicle consists of epithelial cells that form a closed ellipsoid containing a fluid-filled lumen inside. This lumen is not a cell, so we manually relabelled the area as the background.

Conclusion

Existing 3D cell segmentation methods face two challenges: (1) low accuracy in the presence of densely packed cells and (2) low robustness due to the need to fine-tune multiple manually selected hyperparameters on new datasets. This study aims to tackle these challenges by developing a deep learning-based two-stage pipeline, 3DCellSeg, for accurate and robust 3D cell segmentation with high computational efficiency. Specifically, the first stage utilizes a light-weight CNN model, 3DCellSegNet, to output the voxel-wise foreground, background, and membrane masks. A novel loss function, 3DCellSeg Loss, is incorporated into the model training process to address the clumped cell challenge. The second stage, which does not require any parameters from the first stage, pre-clusters the labeled voxels into super voxels using a standard watershed algorithm. A novel touching area-based clustering algorithm, TASCAN, is adopted to assemble the super voxels into cell instances while fine-tuning only one hyperparameter, i.e., the minimum touching area between two super voxels of the cell foreground, to better separate the clumped cells among the foreground voxel clusters.

Our experiments on four animal and plant datasets, namely, HMS (animal), ATAS (plant), LRP (plant), and Ovules (plant), show that 3DCellSeg has outperformed the state-of-the-art models on HMS, ATAS, and LRP, achieving an overall accuracy of 76.4%, 95.6%, and 74.7%, respectively. 3DCellSeg has also reached comparable performance to the state-of-the-art results from Ovules, achieving an overall accuracy of 82.2%. Our ablation studies further reveal that 3DCellSeg's improvement in performance is attributable to 3DCellSeg Loss, 3DCellSegNet, and TASCAN, while showing why the whole 3DCellSeg pipeline is more robust across the four datasets. First, the use of 3DCellSeg Loss, a tailored loss function for cell segmentation, has improved the accuracy of voxel classification, especially for more irregular cells presented in the LRP and Ovules datasets. Second, 3DCellSegNet, a light-weight CNN structure using few parameters for cell segmentation, is more accurate and efficient than more complex CNN models when using different datasets with limited training samples. Third, TASCAN is slightly more accurate than its counterpart, is more robust and computationally efficient, requiring only one hyperparameter. Finally, after re-training 3DCellSegNet using a few samples from a different dataset, 3DCellSeg still performs well, demonstrating that our model is robust and transferable to new datasets. The experimental results and the ablation studies suggest that our novel 3DCellSeg can advance research on 3D instance segmentation; it can serve a powerful cell-based disease identification tool, such as cancer diagnostics, when our cell segmentation model is further trained on labelled human cancer/normal cell images. In the future, we will develop an end-to-end deep learning pipeline to segment cell instances in 3D images directly. We will also incorporate more domain-specific knowledge related to the cell characteristics into the pipeline to improve performance.

Data availability

The code is available on <https://github.com/AntonotnaWang/3DCellSeg>. The HMS dataset used during this study is available upon email request to the author Kishore R. Mosaliganti. The ATAS, LRP, and Ovules datasets used during this study are available from the previous studies cited in this article (Willis et al., 2016; Barro et al. 2019; Tofanelli et al., 2019).

Received: 24 July 2021; Accepted: 9 December 2021

Published online: 10 January 2022

References

- Shostak, S. Histology's nomenclature: past, present and future. *Bio. Syst.* **2**, 122 (2013).
- Usaj, M. M. *et al.* High-content screening for quantitative cell biology. *Trends Cell Biol.* **26**(8), 598–611 (2016).
- Williams, S., Henricks, W. H., Becich, M. J., Toscano, M. & Carter, A. B. Telepathology for patient care: what am I getting myself into?. *Adv. Anat. Pathol.* **17**(2), 130–149 (2010).
- Dey, P. Cancer nucleus: morphology and beyond. *Diagn. Cytopathol.* **38**(5), 382–390 (2010).
- Veta, M., Pluim, J. P. W., Van Diest, P. J. & Viergever, M. A. Breast cancer histopathology image analysis: a review. *IEEE Trans. Biomed. Eng.* **61**(5), 1400–1411 (2014).
- S.-H. Chen *et al.*, Altered peripheral profile of blood cells in Alzheimer disease: a hospital-based case-control study. *Med. (Baltim.)* **96**(21) 2017.
- Longo, D. & Hasty, J. Dynamics of single-cell gene expression. *Mol. Syst. Biol.* **2**(1), 64 (2006).
- Megason, S. G. & Fraser, S. E. Imaging in systems biology. *Cell* **130**(5), 784–795 (2007).
- Falk, T. *et al.* U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* **16**(1), 67–70 (2019).
- Van Valen, D. A. *et al.* Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Comput. Biol.* **12**(11), e1005177 (2016).
- Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, 424–432 (2016).
- Milletari, F., Navab, N., & Ahmadi, S. A. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *Proc. - 2016 4th International. Conference 3D Vision, 3DV 2016*, 565–571 (2016).
- Chen, H., Dou, Q., Yu, L., Qin, J. & Heng, P.-A. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *Neuroimage* **170**, 446–455 (2018).
- Dou, Q. *et al.* 3D deeply supervised network for automated segmentation of volumetric medical images. *Med. Image Anal.* **41**, 40–54 (2017).
- Guay, M. D., Emam, Z. A. S., Anderson, A. B., Aronova, M. A., & Leapman, R. D. Dense cellular segmentation for EM using 2D-3D neural network ensembles. *BioRxiv*, (2020).
- Q. Yu *et al.*, C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4126–4135 (2020).
- Xu, J., Li, M., & Zhu, Z. Automatic data augmentation for 3D medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 378–387 (2020).
- Chen, H., Qi, X., Yu, L., & Heng, P.-A. DCAN: deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2487–2496 (2016).
- Bai, M., & Urtasun, R. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5221–5229 (2017).
- Jiang, J., Kao, P.-Y., Belteton, S. A., Szymanski, D. B., & Manjunath, B. S. Accurate 3D cell segmentation using deep features and CRF refinement. In *2019 IEEE International Conference on Image Processing (ICIP)*, 1555–1559 (2019).
- Eschweiler, D., Spina, T. V., Choudhury, R. C., Meyerowitz, E., Cunha, A., & Stegmaier, J. CNN-based preprocessing to optimize watershed-based cell segmentation in 3D confocal microscopy images. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, 223–227 (2019).
- Wolny, A. *et al.* Accurate and versatile 3D segmentation of plant tissues at cellular resolution. *Elife* **9**, 1–34 (2020).
- Kirschbaum, E., Bailoni, A., & Hamprecht, F. A. DISCo: deep learning, instance segmentation, and correlations for cell segmentation in calcium imaging. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 151–162 (2020).
- Scherr, T., Löffler, K., Böhland, M. & Mikut, R. Cell segmentation and tracking using CNN-based distance predictions and a graph-based matching strategy. *PLoS ONE* **15**(12), e0243219 (2020).
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988 (2017).
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587 (2014).
- Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99 (2015).
- He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn, In *Proceedings of the IEEE international conference on computer vision*, 2961–2969 (2017).
- J. Yi *et al.*, Multi-scale cell instance segmentation with keypoint graph based bounding boxes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 369–377 (2019).
- Qi, L. *et al.*, Pointnets: Point-based instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021).
- Tian, Z., Shen, C., Chen, H., & He, T. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9627–9636 (2019).
- Lee, Y. & Park, J. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13906–13915 (2020).
- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9157–9166 (2019).
- Jaeger, P. F. *et al.*, Retina U-Net: Embarrassingly simple exploitation of segmentation supervision for medical object detection. In *Machine Learning for Health Workshop*, 171–183 (2020).
- Zhao, Z., Yang, L., Zheng, H., Guldner, I. H., Zhang, S. & Chen, D. Z. Deep learning based instance segmentation in 3D biomedical images using weak annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 352–360 (2018).
- Mahmood, F. *et al.*, Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *IEEE Trans. Med. Imaging*, 1–1, (2019).
- Payer, C., Štern, D., Neff, T., Bischof, H. & Urschler, M. Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 3–11 (2018).
- Y. Xiang, C. Xie, A. Mousavian, and D. Fox, “Learning RGB-D feature embeddings for unseen object instance segmentation,” *arXiv Prepr. arXiv2007.15157*, 2020.
- Schmidt, U., Weigert, M., Broaddus, C., Myers, G. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 265–273 (2018).
- Chen, X., Girshick, R., He, K. & Dollár, P. Tensormask: A foundation for dense object segmentation. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2061–2069 (2019).
- Sofiuk, K., Barinova, O., & Konushin, A. Adaptis: Adaptive instance selection network. In *Proc. of the IEEE/CVF International Conference on Computer Vision*, 7355–7363 (2019).
- Tian, Z., Shen, C. & Chen, H. Conditional convolutions for instance segmentation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, 282–298 (2020).

43. Weigert, M., Schmidt, U., Haase, R., Sugawara, K., & Myers, G. Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3666–3673 (2020).
44. Takko, H. *et al.* ShapeMetrics: A userfriendly pipeline for 3D cell segmentation and spatial tissue analysis. *Dev. Biol.* **462**(1), 7–19 (2020).
45. Eschweiler, D., Rethwisch, M., Koppers, S., & Stegmaier, J. Spherical harmonics for shape-constrained 3D cell segmentation. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 792–796 (2021).
46. Bailoni, A., Pape, C., Wolf, S., Beier, T., Kreshuk, A. & Hamprecht, F. A. A generalized framework for agglomerative clustering of signed graphs applied to instance segmentation. *arXiv*, 1–19 (2019).
47. Wolf, S. *et al.*, The mutex watershed: efficient, parameter-free image partitioning. In *Proc. of the European Conference on Computer Vision (ECCV)*, 546–562 (2018).
48. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **96**(34), 226–231 (1996).
49. Willis, L. *et al.* Cell size and growth regulation in the Arabidopsis thaliana apical stem cell niche. *Proc. Natl. Acad. Sci.* **113**(51), E8238–E8246 (2016).
50. Barro, A. V. *et al.* Cytoskeleton dynamics are necessary for early events of lateral root initiation in Arabidopsis. *Curr. Biol.* **29**(15), 2443–2454 (2019).
51. Tofanelli, R., Vijayan, A., Scholz, S. & Schneitz, K. Protocol for rapid clearing and staining of fixed Arabidopsis ovules for improved imaging by confocal laser scanning microscopy. *Plant Methods* **15**(1), 1–13 (2019).
52. Arganda-Carreras, I. *et al.* Crowdsourcing the creation of image segmentation algorithms for connectomics. *Front. Neuroanat.* **9**, 142 (2015).
53. Meilã, M. Comparing clusterings: an axiomatic view. In *Proc. of the 22nd International Conference on Machine learning*, 577–584 (2005).
54. Kappes, J. H., Speth, M., Andres, B., Reinelt, G., & Schn, C. Globally optimal image partitioning by multicuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 31–44 (2011).
55. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
56. Li, Y. *et al.*, Fully convolutional networks for panoptic segmentation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 214–223 (2021).
57. Kimmel, R., Kiryati, N. & Bruckstein, A. M. Sub-pixel distance maps and weighted distance transforms. *J. Math. Imaging Vis.* **6**(2), 223–233 (1996).
58. Paszke, A. *et al.*, Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 8026–8037 (2019).
59. Amenta, N., Choi, S. & Kolluri, R. K. The power crust. In *Proc. of the sixth ACM symposium on Solid modeling and applications*, 249–266 (2001)

Acknowledgements

We thank Mr. Kelvin Chan, Senior Research Assistant, the Department of Electrical and Electronic Engineering, the University of Hong Kong, for carefully editing this manuscript.

Author contributions

J.C.L. and V.O.L. were responsible for conceptualization and initial framework development. K.R.M. collected the data. Q.Z. conducted a literature review. Q.Z. and A.W. developed the methodology. A.W. wrote and ran the code for the algorithm. Q.Z., A.W., and Y.H. interpreted the results and wrote the main manuscript. J.C.L. and V.O.L. substantially revised the manuscript. J.C.L., V.O.L., S.M., and S.H. provided crucial guidance on the presentation structure, methodology and analysis. J.C.L. and V.O.L. applied for funding. A.W. and Q.Z. contributed equally.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-021-04048-3>.

Correspondence and requests for materials should be addressed to J.C.K.L. or V.O.K.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022