



3D Nucleus Instance Segmentation for Whole-Brain Microscopy Images

Junbo Ma¹, Oleh Krupa^{2,3}, Madison Rose Glass^{2,3}, Carolyn M. McCormick^{2,3},
David Borland⁴, Minjeong Kim⁵, Jason L. Stein^{2,3}, and Guorong Wu^{1,6}✉

¹ Department of Psychiatry, The University of North Carolina At Chapel Hill, Chapel Hill,
NC 27514, USA

guorong_wu@med.unc.edu

² UNC Neuroscience Center, The University of North Carolina At Chapel Hill, Chapel Hill, NC
27599, USA

³ Department of Genetics, The University of North Carolina At Chapel Hill, Chapel Hill,
NC 27599, USA

⁴ RENCI, The University of North Carolina At Chapel Hill, Chapel Hill, NC 27599, USA

⁵ Department of Computer Science, The University of North Carolina At Greensboro,
Greensboro, NC 27412, USA

⁶ Department of Computer Science, The University of North Carolina
At Chapel Hill, Chapel Hill, NC 27514, USA

Abstract. Tissue clearing and light-sheet microscopy technologies offer new opportunities to quantify the three-dimensional (3D) neural structure at a cellular or even sub-cellular resolution. Although many efforts have been made to recognize nuclei in 3D using deep learning techniques, current state-of-the-art approaches often work in a two-step manner, i.e., first segment nucleus regions within a 2D optical slice and then assemble the regions into the 3D instance of a nucleus. Due to the poor inter-slice resolution in many volumetric microscopy images and lack of contextual information across image slices, the current two-step approaches yield less accurate instance segmentation results. To address these limitations, a novel neural network for 3D nucleus *instance* segmentation (NIS) is proposed, called NIS-Net, which jointly segments and assembles the 3D instances of nuclei. Specifically, a pretext task is designed to predict the image appearance of the to-be-processed slice using the learned context from the processed slices, where the well-characterized contextual information is leveraged to guide the assembly of 3D nuclei instances. Since our NIS-Net progressively identifies nuclei instances by sliding over the entire image stack, our method is capable of segmenting nuclei instances for the whole mouse brain. Experimental results show that our proposed NIS-Net achieves higher accuracy and more reasonable nuclei instances than the current counterpart methods.

Keywords: Nucleus instance segmentation · Light-sheet microscopy images · Self-supervised learning · Variational autoencoder · Deep learning

1 Introduction

Recent advances in tissue clearing and light-sheet microscopy (TCLSM) technologies enable the acquisition of cellular resolution images of intact whole brain samples [12]. This is a great boon to brain structure studies since TCLSM enables us to characterize the three-dimensional (3D) arrangement of immunolabeled cells rather than slices limited to specific brain regions [16]. However, quantitatively analyzing these large-scale microscopy images has lagged behind the ability to acquire them, resulting in their most common use as visualizations rather than quantified results.

The most basic step in the quantification of brain structure involves counting nuclei in defined brain regions, co-localization with immunolabeled cell-type-specific markers, and comparison between groups defined by a biological condition, such as genotype [9, 14]. In this regard, a high-throughput computational method to delineate each nucleus in large-scale microscopy images is of high demand in the neuroscience field. This process includes two key steps: (1) segmentation of nucleus regions in each slice and (2) assembly of nucleus regions into the instance of the nucleus in 3D.

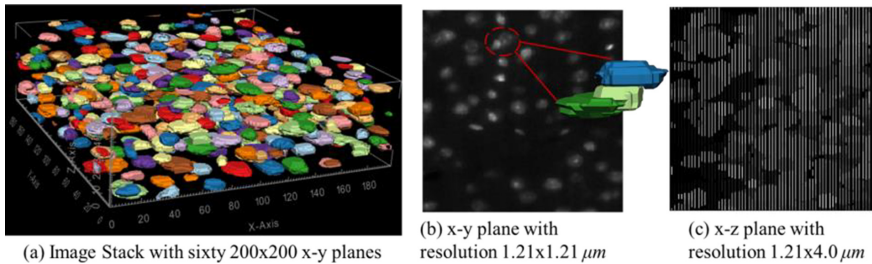


Fig. 1. Typical tissue clearing and light-sheet microscopy images. Nucleus instance segmentation (NIS) aims to detecting and delineating each distinct 3D nucleus at the voxel level. (a) shows the 3D view of an image stack with manually annotated 3D nuclei masks. It has 60 image slices (z-axis), and each slice is a 200×200 pixel (x-y plane) gray-scale image. The color in (a) implies the unique identity of each nucleus. The image resolution is $1.21 \times 1.21 \times 4.0 \mu\text{m}$. (b) shows a typical image slice (x-y plane) from the 3D image stack. The red circle shows an example of touching nuclei. (c) shows the slice-by-slice nature of the image stack in the x-z plane, where image artifacts across slices create challenges for 3D nucleus instance segmentation.

Regarding nucleus region segmentation, a plethora of efforts has been made to improve the accuracy and scalability of segmentation results, from classic deformable models [11] to recent deep learning techniques such as convolutional neural network (CNN) based approaches [7, 15, 18]. Since it is time-consuming and extremely laborious to have a large pool of manual annotations of 3D nuclei, the majority of current methods segment nucleus regions using 2D methods. To alleviate this issue, some works proposed to train the neural network with synthetic 3D nuclei samples [2]. However, the synthetic samples are biased on the synthesizing dataset and may not generalize well in real applications.

Although state-of-the-art deep learning techniques, such as 3D U-Net [15] and 3D Mask R-CNN [4], achieve great success in computer vision and medical imaging

areas, they often yield less-accurate nucleus instance segmentation results in microscopy images due to the following two reasons. **(1) Anisotropic Image Resolution.** Since light-sheet microscopy images are acquired slice by slice, the intra-slice resolution (the x-y plane in Fig. 1(b)) is usually 2–10 times higher than the inter-slice resolution (the x-z plane in Fig. 1(c)), which leads to a significant amount of image artifacts such as discontinuous appearance and spurious boundaries. As shown in Fig. 1(a)(b) and Fig. 3(a), the large inter-slice gap leads to irregular and highly distorted 3D shapes of the nuclei. Since conventional deep neural networks for 3D nucleus region segmentation work under the assumption that image appearance is spatially smooth and image resolution is isotropic, their segmentation results are often sub-optimal. **(2) Densely Packed Nuclei.** As shown in Fig. 1(b), it is common that nuclei touch each other in certain cell-dense regions of the brain, such as the hippocampus. Disentangling individual nuclei within cell-dense regions is important in many neuroscience applications. However, it is very challenging to address this issue, partially due to the lack of 3D manual annotations of touching nuclei.

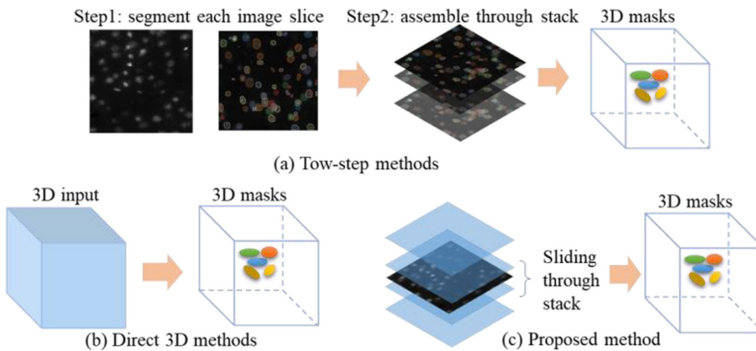


Fig. 2. A high-level sketch of various solutions for 3D nuclei instance segmentation. (a) Two-step methods first segment the 2D nucleus regions in each image slice, then assemble the 2D regions into 3D masks. (b) The end-to-end solution directly predicts 3D masks of all possible nucleus instances. (c) Our proposed method jointly segments nuclear regions and assembles the nuclear voxels into individual 3D nuclei instances by sliding over the image stack.

Regarding nucleus instance assembly, most of the current methods first segment the 2D masks of nuclei in each slice, then deploy a post hoc image processing algorithm such as watershed or region growing to delineate the 3D contour for each nucleus from the segmented 2D masks [7, 18]. However, due to the challenges shown in Fig. 3, it is difficult to assemble the nucleus regions into the 3D instance of nuclei without knowing the 3D context of the nuclei-to-nuclei relationship.

Figure 2 illustrates a high-level sketch of different strategies for 3D nuclei instance segmentation. Due to the large inter-slice gap and complex layout of nuclei in the 3D environment, conventional two-step methods Fig. 2(a) have very limited power to accurately assemble all identified nucleus regions at the voxel level to the nuclei instances. End-to-end direct 3D methods Fig. 2(b) seem like the ultimate solution for recognizing nuclei instances from a given image stack. However, the spatial relationships between

nuclei in the image stack are highly complicated, not to mention the scenario of densely packed nuclei. Such challenges make the learning complexity in the end-to-end approach grow exponentially as the size of the image stack increases.

In contrast, our NIS-Net Fig. 2(c) segments 3D nucleus instances by progressively considering part of the 3D image stack in a sliding window manner and jointly segments and assembles 3D masks of nucleus instances throughout the sliding process. In this way, our NIS-Net avoids the limitations of either two-step or end-to-end methods and achieves comparable performance to human experts with very limited annotated training samples. Furthermore, our NIS-Net provides a flexible end-to-end 3D solution that is less demanding on 3D manual annotations but can be scaled up to whole-brain ($> 10^5$ voxels in each dimension) nucleus instance segmentation. We have evaluated the nucleus instance segmentation results from our NIS-Net on light-sheet microscopy images of the mouse brain. Compared to current state-of-the-art methods in Sect. 4, our NIS-Net achieves more accurate and reasonable 3D nuclei identification results, which indicates its applicability for neuroscience applications.

2 Problem Formulation

The 3D nucleus instance segmentation problem can be formally defined as follows: Given a 3D TCLSM image stack V that has T slices, $V = \{S_1, \dots, S_T\}$, S_t is the t -th slice in the image stack V . The aim of NIS is to segment all the nucleus instances at the voxel level in the image stack. Suppose there are N nucleus instances in the image stack V . In an image slice S_t , we denote the 2D segmentation masks $M_t = \{m_{n,t}\}$ and their corresponding bounding boxes $B_t = \{b_{n,t}\}$. Segmentation mask $m_{n,t}$ is a binary representation of the nuclear regions in each image slice, where 1 means the corresponding pixel belonging to nucleus instance n . $id_{n,t}$ is the unique identity corresponding to $m_{n,t}$. Then, the segmentation masks in M with the same identity across the image stack can be assembled as the 3D mask of a nucleus.

There are three major challenges to this problem. The first one is the issue of the appearance gap between two consecutive slices. For example, in Fig. 3(a), the appearance gap leads to a significant shape change of the 2D masks from the same nucleus between adjacent slices. Also, from Fig. 3(a), we can see that the masks of the same nucleus only appear in very few slices. This is because the size of the nucleus is small compared to the physical distance between two image slices when acquiring the image stack. In our dataset, one nucleus usually lies across 5.4 slices on average, making the appearance gap issue much more challenging to solve. The second challenge is how to identify the segmentation mask belonging to the same nucleus at the voxel level. As shown in Fig. 3(b), there are 68 nucleus instances in this 200×200 image slice. Moreover, the instances crowded and touched each other in some areas, which makes it much harder to determine the mask boundaries between the touching nuclei instances, such as the red rectangle in Fig. 3(b) shows. The third challenge is that 3D shapes are highly diverse in the image stack, as the nuclei are flexible to change shape when crowding together. As shown in Fig. 3(c), these nuclei not only crowd within a slice but also across slices. Thus, the 3D shape of nuclei can change dramatically between slices when crowding together, which leads to more complex 3D structures and makes NIS more challenging

to solve. An additional possible complication is a rare situation where one nucleus may have two or more non-connected segmentation masks in one image slice. Such a rare case may happen when a nucleus bends in 3D like a banana while the plane of an image slice coincidentally cuts the nucleus through its two ends without cutting the “body.” In our current implementation, we assume one nucleus only has one segmentation mask in an image slice.

All these challenges make 3D NIS a unique problem. To address these challenges, a novel NIS-Net model is proposed in the next section.

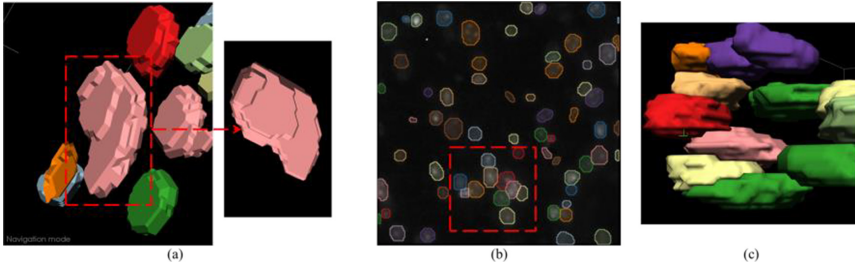


Fig. 3. Three challenging scenarios. (a) shows a nucleus’s 2D mask shape changing dramatically through consecutive image slices. Also, we can see that nuclei only lie across very few slices. (b) shows the density of nuclei in one image slice. In the red rectangle area, nuclei crowded together. (c) shows the nuclei crowding and touching each other in 3D, making the NIS more challenging to solve.

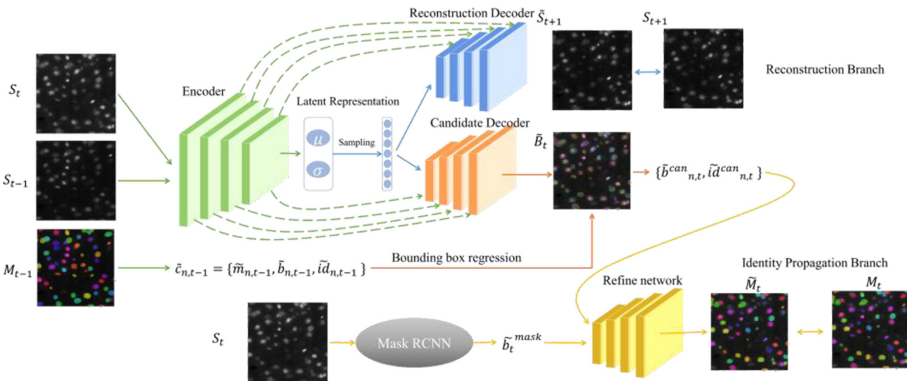


Fig. 4. The architecture of the proposed NIS-Net. It consists of two branches, the reconstruction branch and the identity propagation branch. The reconstruction branch aims to reconstruct the next image slice S_{t+1} , from the previous image slice S_{t-1} and current image slice S_t . This will force the model to encode the 3D context information in the adjacent slices. The identity propagation branch will first propagate the identities of existing nuclei in the previous slice to the current slice, then refine the masks and assign new identities to the newly arising nuclei. Masks with the same identity will be considered as coming from the same nucleus.

3 Proposed Model

As discussed in the previous sections, because of the high density of the instances, the large appearance gap, and the complex 3D structure, the 3D NIS problem is very challenging. To address these challenges, our approach fully explores the 3D context information between adjacent slices. Inspired by the self-supervised learning technique [5, 6, 10], a pretext task is specifically designed to ensure the model learns the 3D context information. The proposed model follows a Variational Autoencoder (VAE) architecture [8] with two branches, the reconstruction branch, and the identity propagation branch, as shown in Fig. 4. We will introduce them in detail in the following sections.

3.1 Encoder

The encoder of NIS-Net utilizes the Residual Neural Network (Res-Net) as the backbone to extract image features. It takes the previous image slice S_{t-1} and current image slice S_t as inputs and encodes them into a latent representation vector. Following the VAE’s architecture, instead of producing a latent representation vector with actual values, the encoder produces a vector of means μ and a vector of standard derivations σ . Then, each element in the latent representation vector is sampled from the (μ, σ) pair. In this way, the generalization ability of the NIS-Net is expanded, and the requirement of a large amount of annotated samples is partially relaxed through the sampling process. However, the VAE itself cannot guarantee to focus on encoding the 3D successive information between slices. Thus, a specially designed pretext learning task is introduced into VAE in the reconstruction branch.

3.2 Reconstruction Branch

A pretext learning task is introduced into this branch, which aims to reconstruct the next image slice S_{t+1} based on the successive information learned from the previous slice S_{t-1} and current slice S_t . To ensure the reconstruction decoder successfully decode the latent representation into the next image slice S_{t+1} , the latent representation must have encoded enough successive 3D information between slices throughout the image stack.

The reconstruction decoder is also implemented with residual blocks. Subpixel convolution [13] is utilized to perform up-sampling between residual blocks. However, in practice, because of the aliasing effect of up-sampling, the reconstructed images are usually very “blurry.” To avoid this, we enhance the reconstruction branch by the spatial information extracted and preserved during the encoding process, which follows the implementation of the U-Net. The latent connections, as the dashed lines show in Fig. 4, can reduce the aliasing effect of up-sampling and enrich the texture details, thus, resulting in sharper reconstructed images. The mean square error (MSE) between the reconstructed image and the ground truth image is used as the loss function for this branch, which means we do not need extra annotation for this task, and this task learns from itself.

By solving this pretext learning task, the proposed NIS-Net can focus on encoding more successive information into the latent representation, and the backbone Res-Net in Encoder can extract more spatially informative features passing through the latent

connections. All this information is used to process the identities propagating between two adjacent slices throughout the image stack.

3.3 Identity Propagation Branch

The identity propagation branch has two steps. The first step aims to propagate existing identities $\{\tilde{id}_{n,t-1}\}$ from the previous image slice S_{t-1} to the current slice S_t , which is performed by the candidate decoder, as shown in Fig. 4. The second step aims to deal with the newly arising nuclei in the current slice that do not exist in previous slices, which is performed by the refine network.

The candidate decoder has a similar implementation to the reconstruction decoder. It first decodes the latent representation learned by the encoder into a latent feature map. Then, following the bounding box regression implementation of Fast-RCNN [3], a Region of Interest Alignment (RoIAlign) network is used to find the most relevant areas. However, instead of relying on the region proposal network to generate the regions of interest (RoIs) on the latent feature map, we use the bounding boxes from the previous slice $B_{t-1} = \{b_{n,t-1}\}$ as the RoI. If a bounding box $\tilde{b}_{n,t}^{can}$ is detected with the region of the bounding box $b_{n,t-1}$, the identity of the detected bounding box $\tilde{id}_{n,t}^{can}$ will inherit the identity of $b_{n,t-1}$. Finally, the candidate decoder will produce a list of predicted bounding boxes and their corresponding identities $\{\tilde{b}_{n,t}^{can}, \tilde{id}_{n,t}^{can}\}$.

As there may be newly arising nuclei in the current slice, a refine network is designed to solve this case. The first question is how to find these newly arisen nuclei. As Mask-RCNN has already achieved great success in the 2D nuclei segmentation task [18], we first run Mask-RCNN on the current slice S_t to produce another list of bounding boxes \tilde{b}_t^{mask} . Note that, since Mask-RCNN works solely on one 2D image slice, there will be no identities associated with these bounding boxes. With these two bounding box lists, $\{\tilde{b}_{n,t}^{can}, \tilde{id}_{n,t}^{can}\}$ and $\{\tilde{b}_{m,t}^{mask}\}$, the refine network first performs a matching between the two lists. The matching is based on the intersection over union (IoU) ratio between two boxes. Two boxes are considered as a match only if their IoU ratio is greater than a threshold τ . The optimal matching pairs set is obtained by the Hungarian algorithm. For the matched pairs, a new bounding box is created by the union of two boxes, and the same identity is assigned as the $\tilde{id}_{n,t}^{can}$. For the unmatched $\tilde{b}_{n,t}^{can}$, it will keep its identity $\tilde{id}_{n,t}^{can}$. For the unmatched $\tilde{b}_{m,t}^{mask}$, a new identity will be assigned to it, and this one is considered as the newly arisen nucleus. Finally, based on these bounding boxes, the pixel-level instance masks are produced following the implementation of the mask branch in Mask-RCNN [4].

The whole NIS-Net is jointly trained, which means that NIS-Net is backpropagated based on the sum of the three loss functions: (1) the MSE of the reconstruction loss, (2) the bounding box regression loss of the candidate decoder, and (3) the mask loss of the refine network. After all the instances are detected in the image stack, a list of masks and their corresponding identities are produced. Then, the 3D masks of the nuclei instances can be automatically assembled based on the instance identities of the 2D masks.

4 Experiments

As there are no publicly available annotated 3D TCLSM image stacks for the 3D NIS problem, we manually annotated 5 image stacks for our experiments. We use the tool proposed in [1] for annotation and 3D visualization. The details of the 5 image stacks can be found in the following section. The 2D quantitative metric follows the traditional mAP@IoU score from the computer vision field, which is the mean of the average precision (mAP) scores for all the segmentation categories above different levels of the Intersection over Union (IoU) value. The IoU value is calculated as the area of overlap divided by the area of union between the predicted masks and the ground truth masks.

4.1 Dataset

Table 1. The details of our manually annotated dataset.

	Stack1	Stack2	Stack3	Stack4	Stack5
2D Masks	4398	4522	2753	3728	3832
Nuclei	786	816	510	729	707

The dataset we are using in this paper is split from a TCLSM image stack of a whole mouse brain. For the acquisition of the TCLSM image stack, 1 mm paraformaldehyde (PFA) fixed sections of adult mice were processed according to the iDISCO + protocol. Sections were stained with nuclei dye To-Pro3 (1:400; Thermo Fisher), washed, and embedded in agarose blocks to maintain structural integrity during mounting. Light-sheet imaging was performed using a LaVision Ultramicroscope II equipped with a 2X MVPLAPO (0.5NA) objective. The light-sheet thickness was set to 4 μm , and the physical resolution of each voxel in the image stack is $1.21 \times 1.21 \times 4 \mu\text{m}$. We chose 5 sub-image stacks from the top and side of the cortex area and manually annotated the 3D nuclei masks. Each image stack has 60 image slices. Each image slice is a 200×200 pixels image. 2D masks and nuclei counts for the 5 annotated image stacks are listed in Table 1.

4.2 Experiment Setup

With only 5 annotated image stacks, it is impossible to train a complex 3D U-net or 3D Mask-RCNN, both of which require a larger dataset. Thus, in this paper, we focus only on comparing the proposed method with the two-step methods. For the 2D semantic segmentation step in each image slice, we specifically compare with the Mask-RCNN, as our proposed model already needs a fine-tuned Mask-RCNN. For the assembling step, we used the solutions of the top 3 winners in the CVPR 2020 multiple objects tracking challenge [19] and reported the best one to compare with our NIS-Net.

The metric for the 2D segmentation step follows the traditional mean average precision score (mAP). However, for the tracking step and the 3D nuclei segmentation,

traditional evaluation metrics, such as the soft multi-object tracking and segmentation accuracy (sMOTSA) [17], do not work well. This is because these metrics are designed for relatively high frame-rate videos. As one instance may stay in the video for dozens of frames, the miss prediction in several frames will not decrease the metric score too much. However, in our dataset, one nucleus only lies in 5.4 consecutive image slices on average. Any miss prediction will significantly decrease the metric score. Furthermore, these metrics are designed with potential occlusions in mind. However, occlusion cannot happen in the NIS problem. If a nucleus stops appearing in one image slice, there will be no other part of this nucleus in future slices. Thus, in our paper, we use the Dice score to evaluate the 3D nuclei masks evaluation.

The Res-Net101 is used as the backbone for both Mask-RCNN and NIS-Net and initialized with the pre-trained weights on ImageNet. The convolution stride is set to 2, and the size of the convolution kernels is set to 3. 5-fold cross-validation is applied to all methods. In each fold, one image stack is left out for validation, and the other 4 image stacks are used for training. The average score is reported in the next section.

4.3 Results

The proposed NIS-Net itself needs a thoroughly fine-tuned Mask-RCNN to help refine the candidate masks. Thus, the first comparison is the mAP score of the predicted 2D masks between the proposed NIS-Net and Mask RCNN, as shown in Table 2. The mAP score of the proposed NIS-Net is consistently better than Mask-RCNN because the refinement network in the NIS-Net can reduce the miss predictions of Mask-RCNN by keeping the unmatched $\tilde{b}_{n,t}^{can}$. The unmatched $\tilde{b}_{n,t}^{can}$ originally come from the bounding box regression based on the bounding box regions in the previous slice. In other words, this advantage comes from the 3D information successively propagating between adjacent slices. The high mAP score at the lower IoU thresholds indicates the correct detection of nuclei. However, the mAP is significantly dropped at IoU 90%. In some image slices, only several predictions can get through the 90% IoU threshold. This is because it is much harder to exactly match the manual annotation at every voxel.

Figure 5 shows the 2D segmentation results of NIS-Net and the Mask-RCNN in the same area comparing with the manually annotated ground truth. In this area, Fig. 5 both NIS-Net and the Mask-RCNN work well in general Fig. 5(a)(c). In the more detailed comparison shown in Fig. 5(d)(e), our NIS-Net works better than Mask-RCNN in terms of IoU with ground truth. This is because our NIS-Net refines the results produced from both Mask-RCNN and our candidate decoder. In the heavily crowded area, Mask-RCNN mis-predicted the three nuclei as one nucleus, whereas our NIS-Net predicted three nuclei correctly. This may be because our NIS-Net can utilize the 3D information successively from the previous slice to untangle the touching pixels.

Figure 6 shows the 3D segmentation results of NIS-Net, the ground truth, and the two-step method in the same area. As we can see, our NIS-Net did a better job matching the ground truth, even in the area exhibiting touching nuclei. As discussed in the previous sections, multiple object tracking methods are designed ideally for high framerate video data and with occlusions in mind; thus, these methods tend to aggressively connect the masks across the image stack. This is why the two-step methods have a low 3D Dice score.

Table 2. The quantitative comparison of the methods

	50%	60%	70%	80%	90%	3D Dice
mAP@IoU						
Mask-RCNN	0.9486 ± 0.0196	0.9230 ± 0.0234	0.8746 ± 0.0296	0.7592 ± 0.0541	0.3275 ± 0.1052	0.6095 ± 0.0714
NIS-Net	0.9615 ± 0.0129	0.9443 ± 0.0140	0.9170 ± 0.0183	0.8392 ± 0.0316	0.4866 ± 0.0675	0.8184 ± 0.0397
NIS-Net (no pretext)	0.9523 ± 0.0168	0.9269 ± 0.0213	0.8848 ± 0.0248	0.7733 ± 0.0435	0.4327 ± 0.0950	0.7349 ± 0.0641

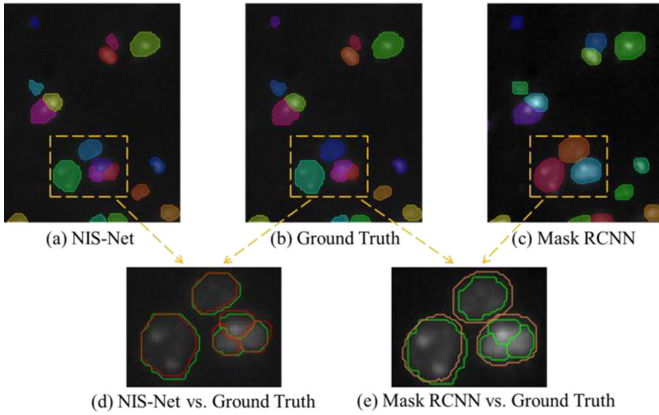


Fig. 5. Sample results of 2D segmentation. (a) (b) (c) shows the sample results of NIS-Net, Ground Truth, and Mask RCNN in the same area, respectively. The colors of masks are randomly generated. (d) shows the detailed comparison. The green contour is the manually annotated ground truth. The red contour is the prediction of NIS-Net. The orange contour is the prediction of Mask-RCNN. It is clear that our NIS-Net works better than Mask RCNN in this area in terms of IoU. Also, Mask RCNN miss-predicted the three crowding nuclei as one.

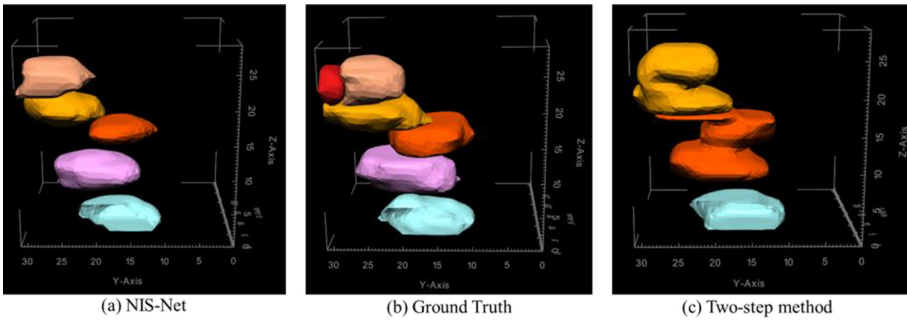


Fig. 6. The 3D results of the same area. Although the proposed method missed predicting one nucleus at the right-above corner, it did a much better job than the two-step method, which tends to connect more masks across the image stack.

Furthermore, we wanted to know how the pretext self-supervised learning task impacts the NIS-Net. Thus, we tested the NIS-Net without the reconstruction decoder. The 2D mAP score and 3D Dice score are reported in Table 2. The quantitative comparison of the methods Without the pretext learning task, the mAP score lies between the full NIS-Net and the Mask-RCNN, while nearer to the results of Mask-RCNN. This result implies that, without the pretext self-supervised learning task, the NIS-Net cannot fully exploit the successive 3D information, and the 2D segmentation power comes more from the Mask-RCNN rather than from the 3D successive information. The significant drop of the 3D Dice score also demonstrates the importance of the pretext self-supervised learning task in the 3D NIS problem.

5 Conclusion

3D nuclei instance segmentation (NIS) for tissue clearing and light-sheet microscopy (TCLSM) image stacks is a challenging problem due to the anisotropic resolution of the TCLSM image stacks, the irregular and highly diverse 3D shapes of the nuclei, and nuclei crowding. In this paper, a novel NIS-Net is proposed to solve this challenging problem via a sliding window methodology, significantly reducing the requirement of annotated 3D samples. Moreover, to fully explore the coherent 3D context information between two successive slices, a pretext learning task designed in a self-supervised learning manner is introduced into the NIS-Net, which enhanced the NIS-Net, focusing on encoding the successive information between adjacent slices. Experimental results demonstrated the effectiveness and efficiency of the proposed NIS-Net. Furthermore, our NIS-Net can be easily scaled up to solve the 3D NIS problem for whole-brain datasets, which we plan to address in our future work.

References

1. Borland, D., et al.: Segmentor: a tool for manual refinement of 3D microscopy annotations. *bioRxiv* (2021). <https://doi.org/10.1101/2021.01.25.428119>
2. Dunn, K.W., et al.: Deepsynth: three-dimensional nuclear segmentation of biological images using neural networks trained with synthetic data. *Sci. Rep.* **9**(1), 1–15 (2019)
3. Girshick, R.: Fast r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448 (2015)
4. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969 (2017)
5. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. In: *Advances in Neural Information Processing Systems*, pp. 15663–15674 (2019)
6. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020)
7. Jung, H., Lodhi, B., Kang, J.: An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images. *BMC Biomed. Eng.* **1**(1), 24 (2019)
8. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: *2nd International Conference on Learning Representations, ICLR2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings* (2014)
9. Lindsey, B.W., Douek, A.M., Loosli, F., Kaslin, J.: A whole brain staining, embedding, and clearing pipeline for adult zebrafish to visualize cell proliferation and morphology in 3-dimensions. *Frontiers Neurosci.* **11**, 750 (2018)
10. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717 (2020)
11. Plissiti, M.E., Nikou, C.: Cell nuclei segmentation by learning a physically based deformable model. In: *2011 17th International Conference on Digital Signal Processing (DSP)*, pp. 1–6. *IEEE* (2011)
12. Richardson, D.S., Lichtman, J.W.: Clarifying tissue clearing. *Cell* **162**(2), 246–257 (2015)
13. Shi, W., et al.: Real-time single image and video super-resolution using an efficient subpixel convolutional neural network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883 (2016)

14. Susaki, E.A., et al.: Whole-brain imaging with single-cell resolution using chemical cocktails and computational analysis. *Cell* **157**(3), 726–739 (2014)
15. Tokuoka, Y., et al.: 3D convolutional neural networks-based segmentation to acquire quantitative criteria of the nucleus during mouse embryogenesis. *NPJ Syst. Biol. Appl.* **6**(1), 1–12 (2020)
16. Vigouroux, R.J., Belle, M., Chédotal, A.: Neuroscience in the third dimension: shedding new light on the brain with tissue clearing. *Mol. Brain* **10**, 33 (2017)
17. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7942–7951 (2019)
18. Zaki, G., et al.: A deep learning pipeline for nucleus segmentation. *Cytometry Part A* **97**(12), 1248–1264 (2020)
19. CVPR 2020 MOTS Challenge. https://motchallenge.net/results/CVPR_2020_MOTS_Challenge/