The 3D Instance Segmentation Network for Synapse Reconstruction from Serial Electron Microscopy Images

Jing Liu\textsuperscript{1,2}, Bei Hong\textsuperscript{1,2}, Qiwei Xie\textsuperscript{3}, Chao Liu\textsuperscript{3}\textsuperscript{*} and Hua Han\textsuperscript{1,2,4}\textsuperscript{*}

\textsuperscript{*}Correspondence: liuchao@bjut.edu.cn; hua.han@ia.ac.cn
\textsuperscript{3}Research Base of Beijing Modern Manufacturing Development, Beijing University of Technology, Beijing, China
\textsuperscript{1}National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
Full list of author information is available at the end of the article

Abstract

Background: The synapse is the key part where neurons communicate with each other. Synaptic plasticity plays a vital role in study and memory. Due to the rapid development of electron microscopy (EM) technology, imaging synapses at nanometer scale possible become possible. However, the automation and effectiveness of the synapse detection algorithm have not yet been satisfactory. The most commonly used method is a two-step solution, where first binary segmentation masks are obtained and then reconstruction results are generated by finding connected components.

Results: In this paper, a novel 3D instance segmentation network which can predict the synapses end to end was proposed. Then, it was also proved that the network can exploit features consistent with the biological structures of synapses by visualizing the network layer. Furthermore, our method was evaluated on two public datasets, and experimental results demonstrated the effectiveness of our proposed method.

Conclusion: The proposed method provided a fast and accurate solution to detecting synapses from serial section EM images. Besides, a block-wise inference strategy which adapts well to large scale EM images was introduced, and it can also help neuroscientists achieve labor-free analysis and quantification of synapses.

Keywords: Synaptic cleft; Instance segmentation; Machine learning; Electron Microscopy

Background

As an important ultrastructure in brains, the synapse has the function of information transmission and storage. Statistics on the size, morphology, distribution, amount and other information of synapses contribute to understanding of the structural information and plasticity analysis on synapses. The synaptic abnormalities associate with many diseases, such as autism and Alzheimer’s disease [1]. Besides, synapses and connected neurons are key components of neural circuits in connectomics. Generally speaking, neurons with different properties and functions are considered as vertices, while the synapses are the connection between two neurons as edges. Research shows that the ratio of the number of synapses to neurons is about one to one thousand [2]. In the face of large amounts of data, fully automated synapse detection and segmentation algorithms are important and necessary.
It is widely known that the synapse is composed of presynaptic structure, postsynaptic structure and the synaptic cleft with the thickness of 30-60nm. Due to the nanometric resolution of synapses, neuroscientists usually observe and image the ultrastructures of synapses with serial section electron microscopy (ssEM). Hence, developing the reliable and precise algorithm to detect synapses from serial EM images has brought about the widespread attention in the field of the computer vision and neuroscience.

Substantial efforts have been recently put into developing specialized algorithms for accurate reconstruction of synapses. Typically, Morales et al. proposed a semi-automated method, which relied on pattern matching and required substantial user interactions [3]. Mishchenko et al. developed a synaptic cleft recognition algorithm to detect postsynaptic densities (PSD) in serial block-face scanning electron microscopy (SBEM) image stacks [4]. Jagadeesh et al. put forward an attribute-based descriptor for characterizing synaptic junctions in rabbits' retinal tissue. Three attribute-based descriptors were designed to represent vesicles, synaptic clefts and ribbons, respectively. Then, they utilized the maximally stable extremal region (MSER) to locate synapses and trained a support vector machine (SVM) to classify the synapses [5]. Navlakha et al. proposed a special tissue staining technique, which highlighted synapses only. Besides, they also obtained synaptic candidate windows through the pre-segmentation result, and then designed a three-part feature for each candidate window. Finally, they built a synaptic classifier by using a traditional machine learning framework [6].

Based on the above reasons, Kreshuk et al. presented a general method of detecting and segmenting synapses in focused ion beam scanning electron microscope (FIB-SEM) image stacks. In this approach, firstly, 38 local texture features were designed for each voxel, and then a random forest classifier was used to predict the category of each voxel as synaptic or non-synaptic type [7]. Inspired by Kreshuk’s method, Becker et al. further developed this approach by designing a local context feature, which can ignore the presence of the asymmetric context information generated by the pre- and post-synaptic regions [8]. As above approaches were only suitable for isotropic data, Kreshuk et al. subsequently proposed an automated approach for synapse segmentation in ssEM images with highly anisotropic axial and lateral resolution. In addition to that, their pipeline consisted of three steps: classification of pixels based on 3D features, segmentation with an Ising model Markov random field (MRF) and classification based on object-level features [9].

Recent years have witnessed the great success of convolutional neural networks (CNNs) in the domain of biocomputing. There is much successful representative work of synapse detection in EM data. Typically, Roncal et al. designed the VESICLE-CNN to segment the synaptic cleft by judging the category of each pixel based on the patch centered on it. The introduction of the CNN has greatly improved the performance at the cost of increased time consumption [10]. Dorkenwald et al. created the SyConn framework, which segmented synaptic connection through a recursive 3D CNN model [11]. Staffler et al. proposed a new approach SynEM for automated detection of synapses from conventionally en-bloc stained EM image stacks. SynEM classified borders between neuronal processes as synaptic or non-synaptic type based on the dense segmentation [12]. Xiao et al. put forward a fully
automated method that relied on deep learning to reconstruct synapses in EM images. The main idea was firstly detecting synapses by the Faster R-CNN [13], and then using GrabCut to segment the synaptic clefts [14].

Although detection and segmentation of synapses in mammal animals have achieved satisfactory results, the performance of these methods applying to insect brain may be not as good. For instance, the synapses in the drosophila brain are hard to identify due to the polyadic nature [15]. It is essential to propose a more general pipeline with stronger characterization ability to detect and segment the synapses from ssEM images.

In this paper, a novel 3D instance segmentation network for synaptic clefts detection and segmentation is introduced, and it can also be trained end to end, thus producing 3D detection and segmentation results in parallel. A block-wise algorithm is developed to achieve accurate detection on large scale datasets. The visualized feature maps from the learned network are consistent with the biology structure of synapses. Besides, the experimental results on two public datasets show promising performances.

** Methods**

**Motivation**

Recently, there are two fundamental ways to reconstruct synapses from EM stacks using the machine learning algorithm. To be specific, one way identifies 3D synapses by classifying the neuronal interfaces of being synaptic or non-synaptic [12, 16], while the other is a two-step method, which firstly obtains binary synaptic masks with the fully convolutional network (FCN) or instance segmentation network, and then detects 3D synapses by finding connected components [11, 14, 17]. It should be noticed that the former method relies heavily on precise neuronal pre-segmentation results. Although it is easy to obtain neural circuits by combining synapses and volume segmentation results, this method is unsuitable for synapse analysis tasks due to the huge computational cost. The two-step method directly predicts synapses by applying CNNs, thus further enhancing the detection performance to a great extent. However, the fundamental disconnect between the mask prediction step and connection step brings the primary disadvantage.

In contrast to the existing methods, a novel 3D instance segmentation network which can simultaneously explore inter- and intraslice features and output 3D synapses in one step was proposed. The differences between the traditional method and our proposed method are illustrated in Figure 1, when Figure 1A depicts the two separated steps to reconstruct synapses in previous methods, while Figure 1B presents the 3D synapses that our proposed network directly predicts.

**Network architecture**

The architecture of our proposed network is illustrated in Figure 2, and concrete parameters of all layers are shown in Table 5. The input of the 3D instance segmentation network is a 3D patch with the size of 512 x 512 x 32, and the outputs are composed of the detection and segmentation results, i.e., 3D bounding boxes and 3D binary masks of detected synapses. With this scheme, the 3d instance masks can be predicted directly from raw image stacks, which means the whole process can be trained end to end.
Specifically, the proposed network contains two-stage detectors and one mask predictor, all of which receive the shared feature maps from the backbone network that is modified based on ResNet50 [18]. Due to the high anisotropy of our test datasets (anisotropic factor is 25 or 10), some parameters of original ResNet50 were revised in order to better fit the data. The kernel size and stride were set as 7 × 7 and 2 × 2 × 1 in the first convolution layer to keep the information along the sz-direction. Then, it is followed by a 3D max-pooling layer with 3 × 3 × 3 pool size and 2 × 2 × 2 stride. Besides, all parameters of the other layers are the same as the original ones, just changing from 2D version to 3D version.

For the first stage detector, it is developed to generate candidate bounding boxes and ensure detection recall rate as much as possible. To locate 3D synapses, two outputs composed of the regression branch and the classification branch for bounding boxes are designed. At each location of the shared feature map, two anchors at different scales, which are 64 × 64 × 6 and 128 × 128 × 6 were predefined respectively, when each anchor has three aspect ratios, corresponding to 1 : 1, 2 : 1 and 1 : 2, respectively. Hence, there are 2 × 3 anchors in each point of feature maps. The classification branch categorizes each predefined anchor as synaptic or non-synaptic one, and then the regression branch regresses the deviation relative to the ground truth. To further optimize the detection precision rate, the top n outputs of the first detector are passed to the second detector. RoI (region of interest) Align layer [19] is utilized to produce fixed-size (7 × 7 × 3) features from shared backbone features. Similar to the first detector, the second detector is designed to perform the classification and regression tasks. In order to feed features to fully connected layers, a 7 × 7 × 3 convolutional layer is first employed to obtain a 1024-D feature. Then, it is followed by two fully connected layers which map the 1024-D to the number of classes (2 in our case) and 2×6-D (the corner number of 3D bounding boxes).

Simultaneously, the mask predictor also takes the RoI features (14 × 14 × 6) as inputs and predicts the segmentation results for every detected RoI. Concretely, the subnetwork contains two 3 × 3 × 3 convolutional layers with the same mode and one 2 × 2 × 1 transposed convolution layer with the stride of 2 × 2 × 1 to preserve the shape of mask (28 × 28 × 6). At last, a 1 × 1 × 1 convolutional layer is applied to map the number of feature channels to the number of classes (2 in our case). For the sake of simplicity, the mask predictor is connected to the RoI Align layer to learn morphological features of synapses. In the phase of inference, it is connected to the second detector, whose detection results are more precise.

Loss function

The loss function of the 3D instance segmentation network consists of three parts, optimizing the first detector, the second detector and the mask predictor, respectively. When W and B are the convolutional kernel set and the bias set, the total
loss function for an image stack can be defined as follows:

\[
\mathcal{L}(\mathcal{W}, \mathcal{B}; y_t, y_p, b_t, b_p, k_t, k_p) = L_{det1} + L_{det2} + L_{mask} \\
= L_{cls1} + L_{cls2} + L_{box1} + L_{box2} + L_{mask} \\
= \frac{1}{N} \sum_{i=1}^{N} -y^1_t \log(y^1_p) + \frac{1}{6N} \sum_{i=1}^{6N} \text{smooth}_{L_1}(b^1_t, b^1_p) \\
+ \frac{1}{n} \sum_{j=1}^{n} -y^2_t \log(y^2_p) + \frac{1}{6n} \sum_{j=1}^{6n} \text{smooth}_{L_1}(b^2_t, b^2_p) \\
+ \frac{1}{nM} \sum_{t=1}^{nM} -k_t \log(k_p) - (1 - k_t) \log(1 - k_p)) \tag{1}
\]

where N, n and M are numbers of total anchors, RoIs and the all pixels of predicted masks, respectively. \(y^1_t\) and \(y^1_p\) represent the ground truth label and the predicted label of classification in the first detector; \(y^2_t\) and \(y^2_p\) refer to the second detector. Similarly, \(b_t\) and \(b_p\) are the ground truth and predicted 3D bounding box \((y, x, z, h, w, c)\), while \(k_t\) and \(k_p\) indicate the ground truth and the predicted 3D masks. Besides, the \(\text{smooth}_{L_1}\) denotes the smooth L1 loss, which can be formulated as:

\[
\text{smooth}_{L_1}(b_t, b_p) = \begin{cases} 
0.5(b_t - b_p)^2 & |b_t - b_p| < 1 \\
|b_t - b_p| - 0.5 & |b_t - b_p| \geq 1
\end{cases} \tag{2}
\]

**Block-wise inference strategy**

Unlike the training process running backward with small patches, the inference process needs to proceed on large scale data. Due to the limited GPU memory, inference runs in a block-wise way. Specifically, the original EM stacks were first decomposed into small blocks \((512 \times 512 \times 32)\) with overlaps of \(128 \times 128 \times 6\) to reduce the boundary effect. This overlap size can cover most of the synapses in our datasets. Then, the trained network predicted the instance segmentation masks of synapses for each block. Next, the small blocks were fused into the stacks of original size. In addition to that, the information of overlapping area was adopted to match neighboring pairs along three directions. For any two areas from neighboring blocks, if the intersection over union (IoU) for both exceeds 0.5, then they were merged as one. Algorithm 2 sketches the pair block matching method.

As shown in Figure 4, firstly, all z-blocks which only recover the scale of z-direction were obtained, and then, all z-blocks were merged into xz-blocks along the x-direction. At last, all xz-blocks were matched successively to produce the final results. The overall block-wise inference strategy on large scale data is described in Algorithm 2.

**Results**

In this section, our proposed network was compared with other state-of-the-art networks on two public datasets to demonstrate the effectiveness. Besides, by visualizing the feature maps of the trained network, it was found that the network can accurately exploit the biological structure characteristics of synapses.
Algorithm 1: A block matching algorithm

Input: Segmentation blocks set \( \{ B_1, B_2, \ldots, B_n \} \), overlap sizes of adjacent blocks \( \{ p_1, p_2, \ldots, p_{n-1} \} \) and the match direction \( d \in \{ x, y, z \} \)

Output: Matched and merged segmentation block \( B \).

1. Initialize: \( block 1 = B_1 \)
2. for \( i = 1 : n - 1 \) do
3. \( block 2 = B_{i+1}, p = p_i \)
4. Extract the overlapping regions \( (overlap1, overlap2) \) according to the overlap size \( p \) and direction \( d \).
5. Compute the volumes of all segmentations in both overlapping regions.
6. for each segmentation \( i \) in \( overlap1 \) do
7. Compute the volume of \( i \).
8. \( v1 = \text{sum}(overlap1 == i) \)
9. for each segmentation \( j \) in \( overlap2 \) do
10. Compute the volume of \( j \).
11. \( v2 = \text{sum}(overlap2 == j) \)
12. Compute the IoU of segmentation \( i \) and segmentation \( j \).
13. \( \text{Ratio}_i = \frac{\text{overlap1} == i \cap \text{overlap2} == j}{\text{overlap1} == i} \)
14. \( \text{Ratio}_j = \frac{\text{overlap1} == i \cap \text{overlap2} == j}{\text{overlap2} == j} \)
15. if \( \text{Ratio}_i > 0.5 \& \text{Ratio}_j > 0.5 \) then
16. Assign label \( i \) to the segmentation \( j \) in \( block2 \).
17. end
18. end
19. Update \( block 1 \) by concatenating the \( block 1 \) and \( block 2 \) along direction \( d \).
20. end

Experimental data and setup

Our proposed method was evaluated on two different datasets (SEM and SNEMI datasets) to prove the generality. The SEM dataset is from the mouse auditory cortex imaged by scanning electron microscopy with a voxel resolution of 2 nm \( \times \) 2 nm \( \times \) 50 nm. Two annotators annotated all the synaptic clefts in 178 slices with cross validation, when each slice had a size of 8576 \( \times \) 7616, and then was split into training and testing datasets. The SNEMI dataset is from a public challenge aiming at automatic segmentation of neurites in 3D, and it also comes from the mouse cortex imaged by scanning electron microscopy with a voxel resolution of 6 nm \( \times \) 6 nm \( \times \) 30 nm, which consists of a training volume and a testing volume with the equal size of 1024 \( \times \) 1024 \( \times \) 100. The EM images and corresponding ground truth are presented in Figure 3.

The network was implemented using Tensorflow and Keras. All training and testing were performed on a server with one NVIDIA Tesla V100 GPU. The network was optimized using stochastic gradient descent (learning rate is 0.001, and the momentum is 0.9). To augment the training dataset as well as avoid over-fitting problem, random rotation\( (90^\circ, 180^\circ, 270^\circ) \), random flipping (up/down, left/right) and adding gaussian noise were employed during the training process.

Biology structure consistent features

To identify a synapse from EM images, three co-occurrence biological features, which are the synaptic vesicles at presynaptic site, the synaptic cleft and the 20-30 nm PSD at postsynaptic site should be satisfied. Recently, researchers have designed hand-crafted features according to the unique biological structures to learn an interface classifier [12], which yields promising performance. Specifically, for
each interface between neurites, 7 subvolumes are defined to indicate the different features in the border and two sides (Figure 5).

Can our proposed 3D instance segmentation network learn the coincident occurrence of regional features? To verify the suspicion, our CNN layer was visualized by adopting the Gradient-weighted Class Activation Mapping (Grad-CAM) [20]. As shown in Figure 6, the heat maps of four sequential layers generated by Grad-CAM are depicted. Results demonstrate that the model identifies synapses by analysing the synaptic cleft, presynaptic and postsynaptic voxels. Coincident occurrence of three unique structures makes a successful detection. Besides, the most important region is the cleft, with importance decreasing gradually from the center outward.

Quantitative comparisons

State-of-the-art methods

The U-Net [21] is a commonly used baseline method in biomedical image segmentation, and there are many improved versions based on the original U-Net, which also achieves good results. The 3D U-Net [22] introduces 3D convolution operations into the original 2D U-Net, which can explore the feature spaces in x-y, x-z and y-z directions, leading to a significant improvement of the learning and fitting ability of the network. Mask R-CNN [19] is the baseline instance segmentation network, which can determine the locations, classes and segmentation masks of all instances in one step. It is easy to extend to other tasks, and has been utilized in synapse segmentation tasks.

Detection performance

To compare with other methods quantitatively, the detection results were evaluated in terms of three fundamental performance indicators, precision and recall. The formulations can be illustrated as follows:

\[
\text{precision} = \frac{TP}{TP + FP} \\
\text{recall} = \frac{TP}{TP + FN} \\
F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

where TP, FP and FN are true positive, false positive and false negative. When evaluating detection performance, a detection box is considered as a sample. If the IoU (intersection over union) between the predicted box and the ground truth box is greater than 23\%, the predicted box is calculated as the positive. The metrics measure the performance on the object-level.

As presented in Table 3 and Table 4, our method achieves the highest precision and F1-score, showing the robustness detection performance. The low recall rate may be caused by the insufficient search space of anchors. To reduce the complexity of the network, only two anchors are defined, and the scale of the third dimension is invariant.
**Segmentation performance**

As for segmentation performance, we don’t adopt the common metrics in the field of semantic segmentation, which is measured by determining whether a voxel is classified correctly. Sizes of synapses are small, and the manual annotations of synapse boundaries are not always accurate. Inspired by the COCO instance segmentation challenge, we still compute the precision and recall metrics. An instance is considered as positive when the segmentation level IoU exceeds 30%. The evaluation is performed in 3D.

As presented in Table 1 and Table 2, our method yields 0.5525 F1-score and 0.3745 F1-score on SEM dataset and SNEMI dataset, which both are the highest scores in all methods. Notably, no post processing is applied for all methods.

**Qualitative comparisons**

To qualitatively illustrate the effectiveness of the proposed method, the results on SEM dataset and SNEMI dataset are shown in Figure 7 and Figure 8, respectively. The ground truth or detected pixels are marked with colors, and the same synapses are labeled with the same color. Due to the imbalance of the positive and negative sample problem, the U-Net predicts few synapses in 2D. Meanwhile, the accuracy of 3D connected components is affected, which causes the poor performances of detection and segmentation. The 3D U-Net exploits the 3D features, which improves the learning ability dramatically. However, the false detection can’t be avoided because of the lack of detection modules. Mask R-CNN achieves low missed detection rate but high false detection rate because the Mask R-CNN only relies on the information of one layer, not the sequential information. Our proposed method combines 3D convolutions into the instance segmentation network, yielding the state-of-the-art performance. 3D synapses are detected and segmented accurately.

**Discussion**

Though our method yields comparable results on the tasks of locating synapses from ssEM images, there is still a lot of room for improvement. In this section, the main advantages and disadvantages are discussed, as well as the future directions in this field.

**Advantages and disadvantages**

In contrast to the two-step methods, our proposed network can output the 3D instance segmentation results of synapses directly without the post-processing of computing connected components. As the process of finding connected components may introduce errors, our end-to-end approach reduces this possibility as much as possible. Besides, our network can learn spatial and temporal features jointly by applying 3D operations including 3D convolution, 3D maxpooling and 3D upsampling. Because of the aligned properties, the 3D contextual information is incredibly necessary for detecting ultrastructures from ssEM images.

The main disadvantage is the high the complexity of the neural network due to the application of 3D convolution and the second detector. The training and testing time increases significantly. Correspondingly, if the memory of the GPU remains the same, the spatial size of inputs generally shrinks to half of the 2D
network, which means the boundary effect becomes more severe since surrounding information lacks. Another problem is the shift error in the segmentation mask caused by the RoI Align layer that adopts bilinear interpolation to replace the original quantization operations, which aims to address the misalignment problem. However, the size of the synapse is very small, and the minor adjustments such as resampling have a huge effect on the final result.

Future directions
To solve the above problems, it is considered that there are three future directions which can further improve the performance. Firstly, the lightweight network can be designed by borrowing the principle of depthwise separable convolution. At present, the fusion strategy of overlapping boundaries is the simple summation. In the future, more complicated and effective algorithm, such as the gaussian-based fusion can be designed. Panoptic segmentation [23] is the hot topic in the field of computer vision. It achieves the tasks of instance segmentation and semantic segmentation in one framework. To overcome the problem of misalignments, the idea of panoptic segmentation can be introduced to avoid the crop operation by combining the fully convolutional network.

Conclusions
In this paper, a novel 3D instance segmentation network is put forward to detect synaptic clefts from ssEM images. The proposed network contains two detectors and one mask predictor, which are adopted to identify the locations and segmentation masks of synapses. The end-to-end training fashion facilitates the fitting capability of the network and significantly enhances the performance. The visualization of the network layer demonstrates that the network can learn features consistent with the unique structure of synapses. Compared with the common two-step method, our approach can generate all 3D synapses in one step. Besides, the experimental results on the public dataset show that our proposed method achieves state-of-the-art performance.

Appendix
Text for this section...

Acknowledgements
The authors would like to thank Mr. Hongtu Ma and colleagues (Institute of Automation, CAS) for the use of the Zeiss Supra55 scanning electron microscope and for their technical support. We also would like to thank Weifu Li and Chi Xiao for providing the SEM dataset.

Funding
This research was supported by the National Natural Science Foundation of China (NOs. 61673381 and 31970960), the Strategic Priority Research Program of Chinese Academy of Science (NO. XDB32030200), Bureau of International Cooperation, CAS (NO. 153D31KYSB20170059).

Abbreviations
EM: Electron microscopy; PSD: Postsynaptic densities; SBEM: Serial block-face scanning electron microscopy; MSER: Maximally stable extremal region; SVM: Support vector machine; FIB-SEM: Focused ion beam scanning electron microscopy; MRF: Markov random field; CNN: Convolutional neural network; FCN: Fully convolutional network; ATUM-SEM: Automated tape-collecting ultramicrotome scanning electron microscopy; IoU: Intersection-over-Union; Grad-CAM: Gradient-weighted Class Activation Mapping;

Availability of data and materials
The data and source code in this paper is available upon request.
Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Consent for publication
Not applicable.

Authors’ contributions
JL designed and conducted the study. JL and BH wrote the manuscript with the help of HH and QX. HH and QX conceived the ideas for all the methods and supervised the experiments.

Author details
1 National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. 2 School of Artificial Intelligence, School of Future Technology, University of Chinese Academy of Sciences, Beijing, China. 3 Research Base of Beijing Modern Manufacturing Development, Beijing University of Technology, Beijing, China. 4 CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China.

References
1. Kanner, L.: Irrelevant and metaphorical language in early infantile autism. American journal of Psychiatry 103(2), 242–246 (1946)
2. Tapia, J.C., Kasthuri, N., Hayworth, K.J., Schalek, R., Lichtman, J.W., Smith, S.J., Buchanan, J.: High-contrast en bloc staining of neuronal tissue for field emission scanning electron microscopy. Nature protocols 7(2), 193 (2012)
3. Morales, J., Alonso-Nanclares, L., Rodríguez, J.-R., DeFelipe, J., Rodríguez, Á., Merchán-Pérez, Á.: Espina: a tool for the automated segmentation and counting of synapses in large stacks of electron microscopy images. Frontiers in neuroanatomy 5, 18 (2011)
4. Mishchenko, Y., Yu, T., Spaceck, J., Mendenhall, J., Harris, K.M., Chklovskii, D.B.: Ultrastructural analysis of hippocampal neuropil from the connectomics perspective. Neuron 67(6), 1009–1020 (2010).
5. Jagadeesh, V., Anderson, J., Jones, B., Marc, R., Fisher, S., Manjunath, B.: Synapse classification and localization in electron micrographs. Pattern Recognition Letters 43, 17–24 (2014)
6. Navlakha, S., Suhan, J., Barth, A.L., Bar-Joseph, Z.: A high-throughput framework to detect synapses in electron microscopy images. Bioinformatics 29(13), 9–17 (2013). doi:10.1093/bioinformatics/btt222.
7. Kreshuk, A., Straehle, C.N., Sommer, C., Koethe, U., Cantoni, M., Knott, G., Hamprecht, F.A.: Automated detection and segmentation of synaptic contacts in nearly isotropic serial electron microscopy images. PLOS ONE 6(10), 1–8 (2011). doi:10.1371/journal.pone.0024899
8. Becker, C., Ali, K., Knott, G., Fua, P.: Learning context cues for synapse segmentation. IEEE transactions on medical imaging 32(10), 1864–1877 (2013)
9. Kreshuk, A., Koethe, U., Pax, E., Bock, D.D., Hamprecht, F.A.: Automated detection of synapses in serial section transmission electron microscopy image stacks. PLOS ONE 9(2), 1–11 (2014). doi:10.1371/journal.pone.0087351
10. Roncal, W.G., Pekala, M., Kaynig-Fittkau, V., Kleissas, D.M., Vogelstein, J.T., Pfister, H., Burns, R., Vogelstein, R.J., Chevillet, M.A., Hager, G.D.: Vesicle: volumetric evaluation of synaptic interfaces using computer vision at large scale. arXiv preprint arXiv:1403.3724 (2014)
11. Dorkenwald, S., Schubert, P.J., Killinger, M.F., Urban, G., Mikula, S., Svarz, F., Kornfeld, J.: Automated synaptic connectivity inference for volume electron microscopy. Nature methods 14(4), 435–442 (2017)
12. Staffler, B., Berning, M., Boergens, K.M., Gour, A., Smagt, P.v.d., Helmstaedter, M.: Synem, automated synapse detection for connectomics. eLife 6, 26414 (2017). doi:10.7554/elife.26414
13. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1137–1149 (2017). doi:10.1109/TPAMI.2016.2577031
14. Xiao, C., Li, W., Deng, H., Chen, X., Yang, Y., Xie, Q., Han, H.: Effective automated pipeline for 3d reconstruction of synapses based on deep learning. BMC bioinformatics 19(1), 263 (2018)
15. Huang, G.B., Scheffer, L.K., Plaza, S.M.: Fully-automatic synapse prediction and validation on a large data set. Frontiers in neural circuits 12, 87 (2018)
16. Motta, A., Berning, M., Boergens, K.M., Staffler, B., Beining, M., Loomba, S., Hennig, P., Wisler, H., Helmstaedter, M.: Dense connectomic reconstruction in layer 4 of the somatosensory cortex. Science 366(6469), 1011.126 (2019). doi:10.1126/science.aaay3134. https://science.sciencemag.org/content/366/6469/aaay3134.full.pdf
17. Dorkenwald, S., Turner, N.L., Macrina, T., Lee, K., Lu, R., Wu, J., Bodor, A.L., Bleckert, A.A., Brittain, D., Kemnitz, N., et al.: Binary and analog variation of synapses between cortical pyramidal neurons. BioRxiv (2019)
18. 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, pp. 770–778 (2016)
19. 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)
20. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, pp. 234–241. Springer, Cham (2015)
22. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: Learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, pp. 424–432. Springer, Cham (2016)
23. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9404–9413 (2019)

**Figures**

**Figure 1** The differences between existing methods (A) and our proposed method (B). In previous methods, firstly, binary masks or 2D instances on each layer are predicted, and then the connected components are found to generate 3D synapses. Our proposed 3D instance segmentation network directly produces 3D instances by exploring the 3D feature space.

**Figure 2** The architecture of the proposed network. The orange and green blocks represent the feature maps of the contracting path and the expansive path in the backbone network, respectively. Blue blocks indicate the feature maps of detectors and mask predictor, while the numbers below the blocks are the shapes of corresponding feature maps.

**Figure 3** The raw images and synaptic clefts in SEM (A) and SNEMI (B) datasets. Different colors indicate different synaptic clefts. Scale bar: 1 µm

**Figure 4** The overall diagram of block-wise fusion strategy. Firstly, the EM image stack is decomposed into small blocks with overlaps. Different colors of blocks indicate different cropping direction. Then, the CNN runs forward to predict 3D synapses for each block. To recover the original scale, the fusion is performed with three steps, corresponding to three axes, respectively. The order is axis-z, axis-x and axis-y. The arrows represent merge directions.

**Figure 5** Subvolumes of hand-crafted features defined in [12]. The regions exactly correspond to presynaptic, cleft and postsynaptic of a synapse. Different colors represent different regions which are used to compute features and related statistics.

**Figure 6** Visualization of our trained network on SEM dataset (A) and SNEMI dataset (B). The top row and the bottom row are the original EM images and corresponding heat maps. Colors indicate the level of network attention.

**Figure 7** Qualitative comparisons with other methods on consecutive four layers from SEM dataset.

**Figure 8** Qualitative comparisons with other methods on consecutive four layers from SNEMI dataset.

**Tables**
Algorithm 2: A block-wise fusion algorithm

\textbf{Input:} Original EM stacks \( I = \{I_1, I_2, \cdots, I_n\} \), block size \((b_x, b_y, b_z)\) and overlap size \( (p_x, p_y, p_z)\). The inputs have \( n \) layers, and the height and width of each layer are \( h \) and \( w \), respectively.

\textbf{Output:} Synapses reconstruction results \( O = \{O_1, O_2, \cdots, O_n\} \). The outputs should contain all 3D synapses of inputs, and the same value indicates the same synapse in 3D, while different values indicate the different synapses.

\begin{verbatim}
1 Initialize: \( c_i = [(h - b_y)/(b_y - p_y)] + 1 \)
2 \( c_j = [(w - b_x)/(b_x - p_x)] + 1 \)
3 \( c_t = [(n - b_z)/(b_z - p_z)] + 1 \)
4 for \( i = 1 : c_i \) do
5   Compute the start point \( \text{start}_i \) and end point \( \text{end}_i \) of \( y \) coordinates.
6   if \( i < c_i - 1 \) then
7     \( \text{start}_i = i \times (b_y - p_y) \)
8     \( \text{overlap}_i = p_y \)
9   else
10      \( \text{start}_i = h - b_y \)
11      \( \text{overlap}_i = (i - 1) \times (b_y - p_y) + b_y - \text{start}_i \)
12   end
13 for \( j = 1 : c_j \) do
14   Compute the start point \( \text{start}_j \) and end point \( \text{end}_j \) of \( x \) coordinates.
15   if \( j < c_j - 1 \) then
16      \( \text{start}_j = j \times (b_x - p_x) \)
17      \( \text{overlap}_j = p_x \)
18   else
19      \( \text{start}_j = w - b_x \)
20      \( \text{overlap}_j = (j - 1) \times (b_x - p_x) + b_x - \text{start}_j \)
21   end
22 for \( t = 1 : c_t \) do
23   Compute the start point \( \text{start}_t \) and end point \( \text{end}_t \) of \( z \) coordinates.
24   if \( t < c_t - 1 \) then
25      \( \text{start}_t = t \times (b_z - p_z) \)
26      \( \text{overlap}_t = p_z \)
27   else
28      \( \text{start}_t = n - b_z \)
29      \( \text{overlap}_t = (t - 1) \times (b_z - p_z) + b_z - \text{start}_t \)
30   end
31 Crop the image block \( b_{ijt} \) from the original stacks according to the coordinates \((\text{start}_i, \text{start}_j, \text{start}_t, \text{end}_i, \text{end}_j, \text{end}_t)\) of 3D bounding box.
32 Obtain the segmentation block \( o_{ijt} \) by feeding \( b_{ijt} \) into 3D instance segmentation network.
33 end
34 Match and fuse the segmentation blocks set \( \{o_{ij,t=1,2, \cdots, c_t}\} \) along \( z \)-axis.
35 \( o_i = \text{block\_match}(\{o_{ij}\}, \{\text{overlap}_{ij}, \text{direction} = z\}) \)
36 end
37 Match and fuse segmentation blocks set \( \{o_{ij,j=1,2, \cdots, c_j}\} \) along \( x \)-axis.
38 \( o_i = \text{block\_match}(\{o_{ij}\}, \{\text{overlap}_{ij}, \text{direction} = x\}) \)
39 end
40 Match and fuse the segmentation blocks set \( \{o_{i,i=1,2, \cdots, c_i}\} \) along \( y \)-axis.
41 \( O = \{O_1, O_2, \cdots, O_n\} = \text{block\_match}(\{o_i\}, \{\text{overlap}_i, \text{direction} = y\}) \)
\end{verbatim}

| Algorithm | Precision | Recall | F1-score |
|-----------|-----------|--------|----------|
| U-Net     | 0.6029    | 0.2091 | 0.3105   |
| 3D U-Net  | 0.7922    | 0.2009 | 0.3205   |
| Mask R-CNN| 0.8682    | 0.3194 | 0.4670   |
| Ours      | 0.6437    | 0.4839 | 0.5525   |

Table 1: Segmentation performances on SEM dataset.

| Algorithm     | Precision | Recall | F1-score |
|---------------|-----------|--------|----------|
| U-Net         | 0.1746    | 0.0476 | 0.0748   |
| 3D U-Net      | 0.6984    | 0.2135 | 0.3271   |
| Mask R-CNN    | 0.6746    | 0.2114 | 0.3219   |
| Ours          | 0.3730    | 0.3760 | 0.3745   |

Table 2: Segmentation performances on SNEMI dataset.
Table 3: Detection performances on SEM dataset.

|                  | Precision | Recall | F1-score |
|------------------|-----------|--------|----------|
| U-Net            | 0.2218    | 0.6586 | 0.3318   |
| 3D U-Net         | 0.2037    | 0.7975 | 0.3245   |
| Mask R-CNN       | 0.3178    | 0.8696 | 0.4654   |
| Ours             | 0.5678    | 0.6535 | 0.6076   |

Table 4: Detection performances on SNEMI dataset.

|                  | Precision | Recall | F1-score |
|------------------|-----------|--------|----------|
| U-Net            | 0.0773    | 0.2093 | 0.1129   |
| 3D U-Net         | 0.1488    | 0.7131 | 0.2462   |
| Mask R-CNN       | 0.2000    | 0.6976 | 0.3108   |
| Ours             | 0.4700    | 0.4200 | 0.4435   |

Table 5: Parameters of all layers in our proposed 3D instance segmentation network.

| Module          | Layer name  | Number of kernels | Kernel size/Stride |
|-----------------|-------------|-------------------|--------------------|
| backbone        | conv1       | 64                | 7 x 7 x 5/2 x 2 x 1 |
|                 | maxpool     | -                 | 3 x 3 x 3/2 x 2 x 2 |
|                 | conv2_x     | 64                | 1 x 1 x 1/1 x 1 x 1 |
|                 |             | 64 x 3            | 3 x 3 x 3/1 x 1 x 1 |
|                 |             | 256               | 1 x 1 x 1/1 x 1 x 1 |
|                 | conv3_x     | 128               | 1 x 1 x 1/1 x 1 x 1 |
|                 |             | 128 x 4           | 3 x 3 x 3/1 x 1 x 1 |
|                 |             | 512               | 1 x 1 x 1/1 x 1 x 1 |
|                 | conv4_x     | 256               | 1 x 1 x 1/2 x 2 x 2 |
|                 |             | 256 x 6           | 3 x 3 x 3/1 x 1 x 1 |
|                 |             | 1024              | 1 x 1 x 1/1 x 1 x 1 |
| First detector  | first_conv1 | 512               | 3 x 3 x 3/1 x 1 x 1 |
|                 | first_class | 6                 | 1 x 1 x 1/1 x 1 x 1 |
|                 | first_box   | 18                | 1 x 1 x 1/1 x 1 x 1 |
| Role features   | roi_align   | -                 | pool_size = 7 x 7 x 3 |
| Second detector | second_conv1| 1024              | 7 x 7 x 3/1 x 1 x 1 |
|                 | second_conv2| 1024              | 1 x 1 x 1/1 x 1 x 1 |
|                 | second_class| 2                 | -                  |
|                 | second_box  | 12                | -                  |
| Mask            | mask_conv1  | 256               | 3 x 3 x 3/1 x 1 x 1 |
|                 | mask_conv2  | 256               | 3 x 3 x 3/1 x 1 x 1 |
|                 | mask_conv3  | 256               | 3 x 3 x 3/1 x 1 x 1 |
|                 | mask_conv4  | 256               | 3 x 3 x 3/1 x 1 x 1 |
|                 | mask_deconv | 256               | 2 x 2 x 2/2 x 2 x 1 |
|                 | mask        | 2                 | 1 x 1 x 1/1 x 1 x 1 |