

Spherical-Patches Extraction for Deep-Learning Based Critical Points Detection in 3D Neuron Microscopy Images

Weixun Chen, Min Liu, Qi Zhan, Yinghui Tan, Erik Meijering, *Fellow, IEEE*, Miroslav Radojević, and Yaonan Wang

Abstract—Digital reconstruction of neuronal structures is very important to neuroscience research. Many existing reconstruction algorithms require a set of good seed points. 3D neuron critical points, including terminations, branch points and cross-over points, are good candidates for such seed points. However, a method that can simultaneously detect all types of critical points has barely been explored.

In this work, we present a method to simultaneously detect all 3 types of 3D critical points in neuron microscopy images, based on a spherical-patches extraction (SPE) method and a 2D multi-stream convolutional neural network (CNN). SPE uses a set of concentric spherical surfaces centered at a given critical point candidate to extract intensity distribution features around the point. Then, a group of 2D spherical patches is generated by projecting the surfaces into 2D rectangular image patches according to the orders of the azimuth and the polar angles. Finally, a 2D multi-stream CNN, in which each stream receives one spherical patch as input, is designed to learn the intensity distribution features from those spherical patches and classify the given critical point candidate into one of four classes: termination, branch point, cross-over point or non-critical point. Experimental results confirm that the proposed method outperforms other state-of-the-art critical points detection methods. The critical points based neuron reconstruction results demonstrate the potential of the detected neuron critical points to be good seed points for neuron reconstruction. Additionally, we have established a public dataset dedicated for neuron critical points detection, which has been released along with this paper.

Index Terms—Critical points detection, neuron reconstruction, deep learning, microscopy images.

I. INTRODUCTION

THE morphology reconstruction of neuronal structures from microscopy images plays a critical role in the understanding of how the brain works [1][2]. The study of 3D neuron morphology allows for the understanding of neuronal outgrowth, regeneration, or networking. This is important to

This work was supported by the National Natural Science Foundation of China under Grant No. 61771189 and the Hunan Provincial Natural Science Foundation of China under Grant No. 2018JJ2060. (*Corresponding author: Min Liu*)

W. Chen, M. Liu, Q. Zhan, Y. Tan, and Y. Wang are with the College of Electrical and Information Engineering and also with National Engineering Laboratory for Robot Vision Perception and Control, Hunan University, Changsha, China. (E-mail: liu_min@hnu.edu.cn).

E. Meijering is with the School of Computer Science and Engineering and the Graduate School of Biomedical Engineering, University of New South Wales, Sydney, Australia. (E-mail: meijering@imagescience.org).

M. Radojević is with the R&D department of Nuctech Netherlands, Rotterdam, The Netherlands.

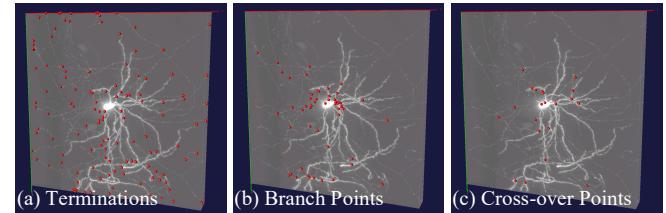


Fig. 1. Examples of critical points of a neuron and its surrounding neuronal structures in a microscopy image of a whole mouse brain. (a) Terminations. (b) Branch points. (c) Cross-over points. Vaa3D [9] was used for visualization.

the research and treatment of central nervous system (CNS) diseases such as Alzheimer's and Parkinson's [3]. Artificial intelligence (AI) techniques, for example artificial neural networks, are also inspired by the interconnectivities of the neurons. Therefore, obtaining a blueprint of the brain's network architecture will also help to the research and development of more advanced AI systems. In order to statistically analyze the neuronal cell and network properties of neuronal structures in microscopy images, reliable digital neuron reconstruction (tracing) of the neuronal structures is required.

In the past few decades, many algorithms and tools have been developed for the digital reconstruction of neuronal structures from microscopy images [4][5][6][13]. Many existing automatic or semi-automatic reconstruction methods rely on the availability and quality of initial seed points. *Terminations, branch points and cross-over points* of neuronal structures, collectively called “critical points” in this work, are good seed points for neuron reconstruction [3][7]. Terminations are the end points of neuron branches, and branch points are defined as the junction points of three neuron branches. Cross-over points are defined as the points of visual intersection of two neuronal structures, falsely suggesting a junction of four or more branches¹. Terminations and branch points are helpful to determine the topology and faithfulness of the reconstruction results [8]. While cross-over points help to identify the potential wrong connections in the automated reconstruction results, which is useful to refine the automated neuron reconstruction results or avoid wrong connections during the reconstruction procedure. In this paper, we aim at simultaneously detecting all 3 types of neuron critical points (see Fig. 1 for examples).

It seems natural to use 3D convolutional neural networks (CNNs) to detect neuron critical points in 3D microscopy images. However, the additional computational complexity (volumetric domain) and data sparsity introduce significant

¹A more detailed illustration of the neuron critical points can be found in Supplementary Material.

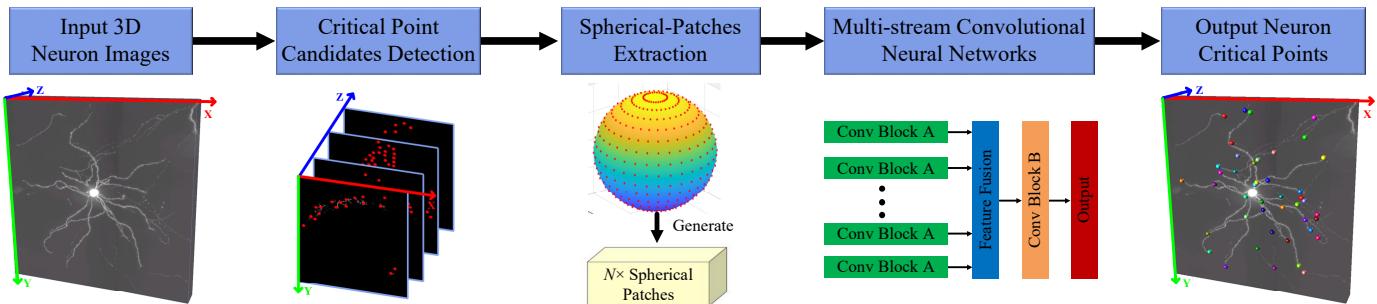


Fig. 2. Overall diagram of the proposed neuron critical points detection method in neuron microscopy images. Given an input 3D neuron image, the first step is to detect a set of critical point candidates in the image. Then, the spherical-patches extraction method is performed at these points to generate spherical patches. Finally, a multi-stream 2D CNN is designed to learn from those spherical patches how to classify the given critical point candidate into termination, branch point, cross-over or non-critical point.

challenges (the ratio of foreground to background of the neurons is less than 5% in the test microscopy images of this work). Therefore, many efforts have been made to deal with volumetric data by using 2D CNNs [7][10][11][12]. Especially for neuron image analysis, the volume of the acquired brain images often contains 20 to 30 or more teravoxels [13], which requires high efficiency for either critical points detection approaches or automatic neuron reconstruction algorithms. Therefore, 2D CNNs with higher efficiency and lower computational cost are much more preferable than 3D CNNs for neuron image analysis.

In this work, we present a method to simultaneously detect all 3 types of 3D critical points in neuron microscopy images, based on the spherical-patches extraction (SPE) method and a 2D multi-stream CNN. SPE uses a set of concentric spherical surfaces, with different radii, centered at a given critical point candidate to extract intensity distribution features around the point. Specifically, each spherical surface is geometrically transformed into a 2D rectangular image patch, whose intensity at any point corresponds to the image intensity on the spherical surface for given azimuth and polar angles. Finally, a 2D multi-stream CNN, in which each stream receives one spherical patch as input, is designed to learn the intensity distribution features from those spherical patches. And the given critical point candidates are classified into termination, branch point, cross-over point or non-critical point. The overall diagram of the proposed method is shown in Fig. 2.

The contribution of this paper mainly resides in the following three aspects:

- The proposed SPE method bridges the gap between 2D networks and 3D neuron critical points detection task without losing 3D structural and contextual information. Many existing methods, for example DeepBranch [7], transform the 3D detection task into 2D by generating multiple views of an object. However, the way they generate the views (e.g., the maximum intensity projections) will compress the structural and contextual information. Therefore, SPE is more robust when handling the dense and complex neuronal structures in microscopy images.
- A multi-stream 2D CNN is designed for the spherical patches generated by SPE. Each stream receives a single spherical patch at the beginning of the network, aiming at fully extracting the multi-scale intensity distribution features from each patch. This allows the network to better identify the interference of the neighboring

neuronal structures and background noises. And a feature fusion module with attention mechanism is designed to aggregate the feature maps of each stream. It can not only make full use of the multi-stream features, but also pay more attention to the contextual information between the streams. To the best of our knowledge, this is the first method to detect all 3 types of 3D critical points in neuron microscopy images simultaneously.

- The proposed method is validated quantitatively on the public BigNeuron dataset [34], and outperforms the compared state-of-the-art methods by about 5% and 1% in average F1-measures for terminations and branch points detection, respectively. Furthermore, we build a more challenging Whole Mouse Brain Sub-image (WMBS) dataset dedicated for neuron critical points detection, including 34 images extracted from a whole mouse brain. Under the same voxel aspect ratio, the z -dimensionality of the BigNeuron images is about 100 voxels, whereas the z -dimensionality of the WMBS images is 500 voxels. Consequently, the WMBS images contain more complicated neuronal structures and are more challenging than the BigNeuron images. The experimental results on the WMBS dataset show that the proposed method outperforms the compared state-of-the-art methods by large margins, about 23% and 27% in average F1-measures for terminations and branch points detection, respectively. In addition, it is a challenging and labor-intensive task to label all the (hundreds of) critical points in an image, because of the ambiguity of neuronal structures. On the one hand, the high degree of mutual occlusion and wiring of neuronal structures require multiple times of image rotation to precisely label a point. On the other hand, the discontinuous or weak signals and the relatively poor z -resolution make it hard to find out and recognize the 3D critical points. In order to advance and benefit automatic whole-brain-scale neuron image analysis, such as 3D neuron critical points detection, seed-based automatic neuron reconstruction and automatic cross-over separation, we have released this dataset along with this paper².

The remainder of this paper is organized as follows: Section II briefly reviews related work, and Section III describes the proposed SPE algorithm and the multi-stream 2D CNN. The

²The dataset is available from <https://github.com/chwx08/WBMS/blob/master/README.md>.

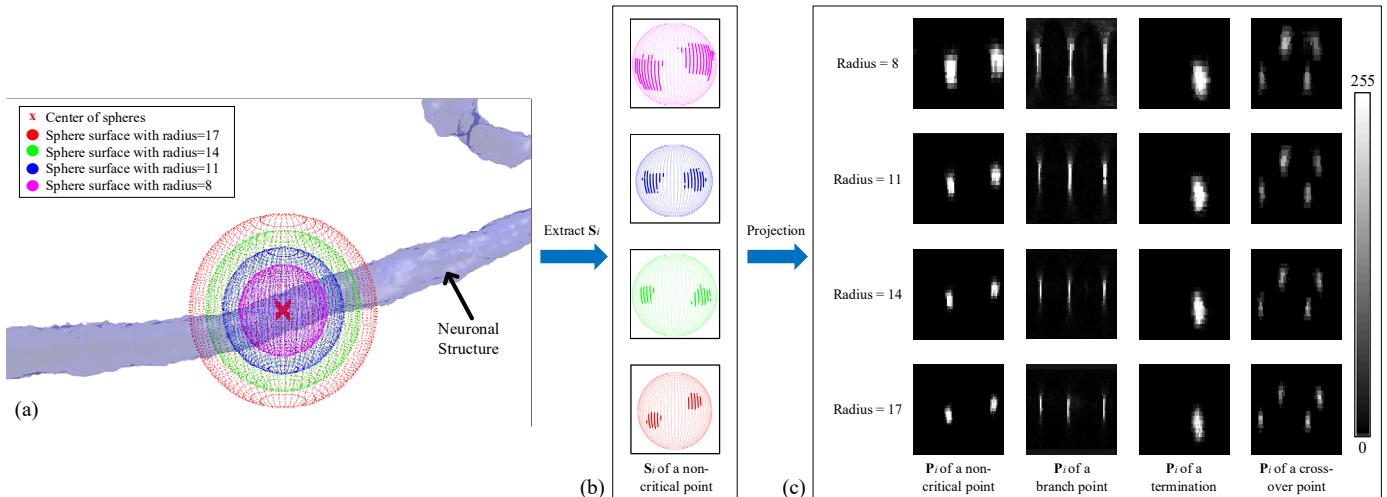


Fig. 3. An overview of SPE. (a) A set of concentric spherical surfaces (different colors indicate different radii) is generated at a non-critical point of neuronal structure in a neuron microscopy image. Only four spheres are generated here, for better view. (b) The intensity distribution features extracted by each spherical surface. The bold regions on the spheres are regions with high intensity values, which indicate the intersection between the spheres and the foreground region of the neuronal structure. (c) Typical spherical patches extracted from a non-critical point, a branch point, a termination and a cross-over point, respectively. It can be clearly seen that the spherical patches of a branch point have three strong responses, whereas those of a termination, a non-critical point and a cross-over point have one, two and four strong responses, respectively.

experiments on neuron microscopy images are presented in Section IV. Finally, we draw our conclusions in Section V.

II. RELATED WORK

Seed-Based Neuron Reconstruction Methods: Many neuron reconstruction methods are initialized by seed points. For example, graph theoretic algorithms have been used to establish connectivity between seeds on or near the centerlines of neuronal structures [14][15]. Methods in [6][16] trace the neuronal structures based on a set of seeds with high ‘tubularity’ value. The method in [17] traces the neuronal structures based on a set of marked point process (MPP) objects on the neuronal structures. An active contour based method is used in [18] to trace the neurons based on seed points along the centerlines. Rivulet [19] and its improved version [20] iteratively trace the neuron fibers back from geodetic furthest points. Methods in [4][21] can trace the neuronal structures with one seed point on the neuron soma or cell body.

Critical Points Detection Methods: Although there are existing methods for termination detection, junction detection and dendritic spines detection of tree-like structures [22][23][24], a method that can simultaneously detect all types of critical points has barely been explored for neuron reconstruction. Neuron Pinpointer [8] was proposed to detect 2D neuron terminations and junctions in fluorescence microscopy images. This algorithm is based on feature extraction and analysis of an angular profile, in combination with a two-stage fuzzy-logic system. A multiscale ray-shooting model [3] was proposed to detect 3D neuron terminations by analyzing the intensity distributions around any termination candidates. DeepBranch [7] was proposed to detect branch points of biomedical images including neuron images, bronchus images and blood vessel images. This two-stage deep-learning based method improves the 3D U-Net [25] to segment the candidate region of branch points and proposes a multi-scale multi-view CNN to detect branch points from the branch point candidates.

III. METHOD

A. Critical Point Candidates Selection

Before using SPE, to reduce computational cost, a set of critical point candidates is selected, containing pixel locations (x, y, z) in an input image I that should be taken into consideration. In this paper, we select the points on the skeleton of the neuronal structures in I as critical point candidates. To extract the skeleton, an improved V-Net [26] designed for neuron microscopy image segmentation is firstly employed to segment I and output a binarized image I_B . The improved V-Net is chosen because of two reasons. Firstly, it is designed for neuron image segmentation and the network architecture is improved to accommodate the unbalanced image sizes in horizontal, vertical and z axes. Secondly, the automatic data labeling method used in this work is able to generate the training samples efficiently. After segmentation, a 3D skeletonization algorithm [27] is employed to find the skeleton of the neurites in I_B . Finally, all the nodes on the skeleton are considered as critical point candidates where the SPE method will be applied. Other methods [28][29][30][31] can also be used to segment the neuronal structures or detect centerline directly. Note that I_B is used only for critical point candidates selection, and we still process the input image I in the remaining steps to avoid possible artifacts introduced by the segmentation step, for example, sharp spurs on the surface of the neuronal structures.

B. Spherical-Patches Extraction (SPE)

The design of SPE is inspired by the elastic boundary projection (EBP) [32] which was proposed for organ segmentation in volumetric medical images. We use the idea of projecting 3D information into 2D rectangle patches, on which 2D networks can be applied. However, EBP cannot be directly used to detect neuron critical points for two reasons. Firstly, EBP only uses one spherical surface to extract the intensity distribution features at each seed point. Due to the large difference between the shape of the organs and neurons,

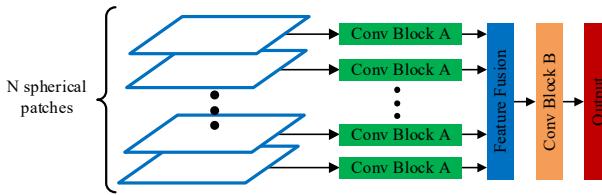


Fig. 4. The overall architecture of the multi-stream CNN for neuron critical points detection. Every single spherical patch is fed to a single stream (*Conv Block A*). Then, the output feature maps of the streams are aggregated by a trainable feature fusion module. The fused feature maps are then fed to *Conv Block B*. Finally, an output layer produces a 4-dimensional vector indicating for each of the four classes the probability that the point belongs to that class.

only one spherical surface is not applicable to detect the critical points in 3D neuron images, because it is not robust to the interference of the background noises and other neighboring neuronal structures. Secondly, EBP iteratively adjusts the spherical surface until it converges to the object boundary. This time-consuming step is not suitable for the large-scale neuron images. Therefore, we improve EBP to SPE by generating a set of spheres with fixed radii at every critical point candidate. The multiple spherical surfaces can extract sufficient intensity distribution features without losing contextual information. This ensures the robustness and performance of our method. And SPE is more efficient than EBP, because there is no need to adjust the spherical surfaces. In the experiments, we demonstrate the effectiveness of SPE by comparing it with other existing patch-extraction methods (Section IV-F). Here we present the SPE method in detail.

Taking a critical point candidate p at location (x, y, z) as a start, the first step of SPE is generating N concentric spherical surfaces, with radii $R = \{r_i | i = 1, \dots, N\}$, centered at p to extract intensity distribution features around it (Fig. 3(a)). In the spherical coordinate system, any point p_m belonging to this set of spherical surfaces can be determined by point p and $(\alpha_{m_1}, \varphi_{m_2}, r_i)$, where $\alpha_{m_1} \in (0, 2\pi]$ is the azimuth angle and $\varphi_{m_2} \in [-\pi/2, \pi/2]$ is the polar angle. In this paper, the azimuth angles are uniformly distributed, i.e., $\alpha_{m_1} = 2\pi m_1 / M_a$, where M_a is the number of azimuth angles and $m_1 = 1, 2, \dots, M_a$. Following the suggestions of the authors in [32], the polar angles have a denser distribution near the equator, i.e., $\varphi_{m_2} = \arccos(2m_2 / (M_p + 1) - 1) - \pi/2$, where M_p is the number of polar angles and $m_2 = 1, 2, \dots, M_p$, so that the points are approximately uniformly distributed over the spherical surface. Given the settings of $\alpha_{m_1}, \varphi_{m_2}$ and R , all the spherical surfaces can be generated.

The second step is using the spherical surfaces to extract the intensity distributions features from a given image I , as shown in Fig. 3(b). As the spherical surfaces are generated in the spherical coordinate system, their coordinates need to be transformed to the Cartesian coordinate system, because the voxel locations in I are indicated by the Cartesian coordinate system. The coordinates transform is,

$$\begin{cases} x_m = x + r_i \cos(\alpha_{m_1}) \cos(\varphi_{m_2}) \\ y_m = y + r_i \sin(\alpha_{m_1}) \cos(\varphi_{m_2}) \\ z_m = z + r_i \sin(\varphi_{m_2}) \end{cases} \quad (1)$$

where (x_m, y_m, z_m) are the Cartesian coordinates of p_m .

The third step is to project the intensity distributions on the spherical surfaces to 2D spherical patches (Fig. 3(c)).

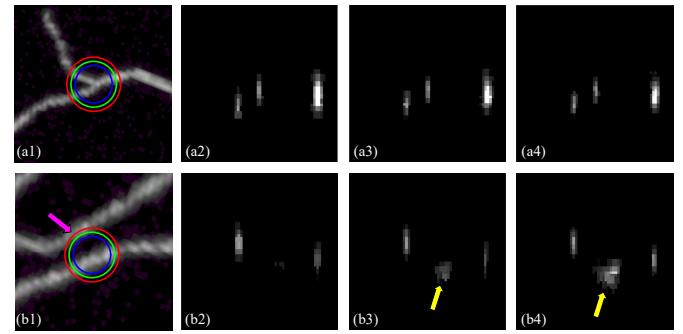


Fig. 5. Illustrations of the spherical patches at different locations. For the clarity of illustration, we show the 2D views of the 3D neuronal structures and use the colored rings to represent the 3D spherical surfaces. (a1) A branch point without other neuronal structures nearby. (a2)-(a4) The spherical patches extracted by the blue, green and red spherical surfaces in (a1), respectively. (b1) Two close neuronal structures. (b2)-(b4) The spherical patches extracted by the blue, green and red spherical surfaces in (b1), respectively. The purple arrow indicates the location where the spherical surface touches the neighboring neuronal structure. The yellow arrows indicate the spots caused by the neighboring neuronal structure.

Specifically, denote S_i by a single spherical surface with radius r_i , the intensity values in S_i will be projected to a 2D rectangle called spherical patch $P_i \in \mathbb{R}^{M_a \times M_p}$, where each row is arranged by the order of α_{m_1} and each column is arranged by the order of φ_{m_2} , i.e., the intensity value at $p_m(\alpha_{m_1}, \varphi_{m_2}, r_i)$ will be projected to the m_1 th row and m_2 th column of P_i . Finally, a total number of N spherical patches $\mathcal{P} = \{P_i | i = 1, \dots, N\}$ are extracted from p .

By using SPE, we transform the intensity distribution of a 3D region of interest into multiple 2D patches which present the intensity distribution in a different way, making the features of different types of critical points more obvious than those in the original image. From Fig. 3(c), we can see that P_i at different types of points will show different patterns. Therefore, 2D CNNs can be used to learn the intensity distribution features from the spherical patches and classify neuron critical points based on the different patterns characterizing different types of points in the patches.

C. 2D Multi-Stream CNN for Neuron Critical Points Detection

In this paper, we design a multi-stream CNN to learn the intensity distribution features from these spherical patches, aiming at classifying any critical point candidate p based on different patterns between different types of critical points. The network architecture is shown in Fig. 4. At the beginning of the multi-stream network, each spherical patch P_i is separately fed to a single stream. The motivation of using the multi-stream setting is that the patches at different scales of the same point may show different patterns, due to the complicated and rich hierarchy of the neuronal structures. And only one stream of the CNN is not effective enough, because all the patches are fused together at the first convolution layer where only low-level features can be extracted.

For further explanation, in Fig. 5(b1)-(b4) we illustrate the patches extracted at the location where two neuronal structures are close to each other. We can observe that the patches in Fig. 5(b3) and (b4) show similar patterns as branch points (i.e., three strong responses), because the spherical surfaces touch the nearby neuronal structure. However, they are still different from the real branch point patches in Fig. 5(a3) and (a4), in

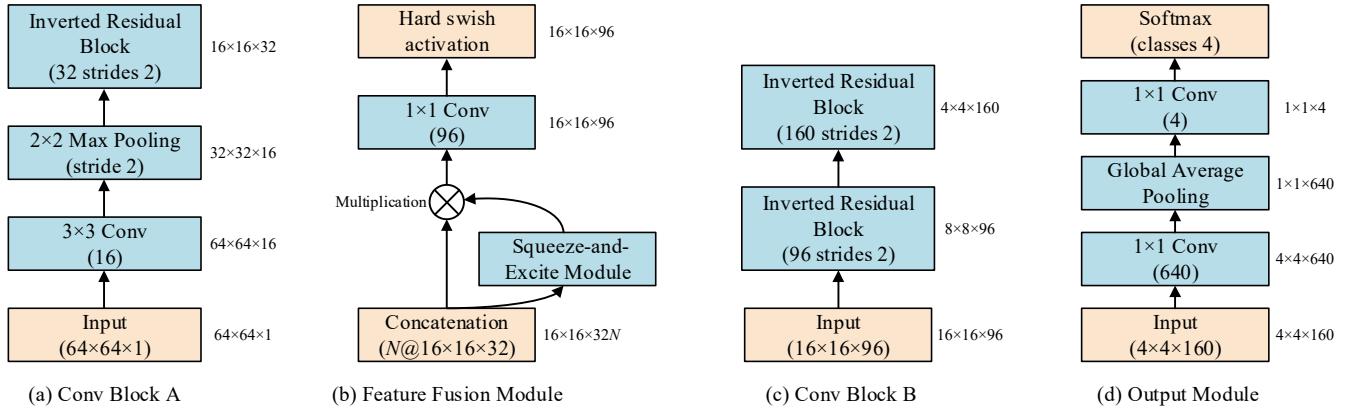


Fig. 6. The schemas for each module in the proposed multi-stream network. Sizes to the right side of each layer summarize the output shape for that layer. The first number at the bottom of the convolution layers and inverted residual blocks indicates the number of filter kernels. “ $N@16 \times 16 \times 32$ ” in (b) means that N feature maps of size $16 \times 16 \times 32$ are concatenated.

terms of the spot shape and the similarity between the adjacent patches. **i)** Spot shape: It can be seen from Fig. 5(a3) that the shapes of the spots in this patch are similar, because they are extracted from the structures of the same branch point. However, the shape of the middle spot, indicated by the yellow arrow in Fig. 5(b3), is quite different from the other spots, because this spot is caused by the neighboring neuronal structure; **ii)** Similarity between the adjacent patches: From Fig. 5(a2)-(a4), we can observe that the patches are similar to each other at a real branch point. However, the patch in Fig. 5(b2) shows a different pattern from the patch in Fig. 5(b3), i.e., a spot suddenly appears in Fig. 5(b3) due to the interference of the neighboring neuronal structure. Consequently, the spot shape and the similarity between the adjacent patches are important features for the network to classify the candidate points. The multi-stream setting enables the network to fully extract the intensity distribution features from each patch. This way, the network can be more sensitive to the spot shape feature than a single-stream network. Furthermore, in order to leverage the patch similarity feature, we design a feature fusion layer with a trainable attention mechanism to aggregate the features extracted by each stream, aiming at making full use of the contextual information between the streams. In the following, we introduce the design of each module. And the schemas for each module are shown in Fig. 6.

Conv Block A: As the network has N streams before being aggregated by the *Feature Fusion Module*, the design of *Conv Block A* should be shallow and lightweight, considering the computational burden and efficiency. Therefore, we use the inverted residual block [33][36] which is a lightweight and effective module.

Feature Fusion Module: The feature maps of each stream will be aggregated into one stream in this module. However, some of the streams may include disturbing information, due to the neighboring neuronal structures or strong background noises, as shown in Fig. 5(b3). Therefore, we perform feature recalibration to selectively weight the importance of the streams, by using the squeeze-and-excite module [37]. It is a trainable mechanism that uses global information to selectively emphasize useful features and suppress less important ones. By using the squeeze-and-excite module, the network learns to leverage the contextual information among

the streams and select the discriminative features. This improves the performance of the network. Since the *Feature Fusion Module* is able to recalibrate all the feature maps generated by the streams, the squeeze-and-excite module is not used in *Conv Block A*, which reduces computational cost. After weighting the importance of the feature maps, we fuse them and reduce the number of feature maps by a 1×1 convolution. At the end of this module, h-swish [36] is used as non-linearity. The default non-linearity of our network is ReLU6 [33], except where stated otherwise.

Conv Block B: In this module, two inverted residual blocks are employed to further extract higher-level features. We implement the squeeze-and-excite module after non-linearity in these inverted residual blocks, which is different from the inverted residual blocks in *Conv Block A*.

Output: We use 1×1 convolution with h-swish non-linearity to expand the feature maps produced by *Conv Block B*. Finally, a global average pooling followed by 1×1 convolution is used to generate a 4-dimentional vector indicating for each of the four classes the probability that the point belongs to that class. Dropout is used after global average pooling, and batch normalization is used after every convolution operation of the network during training.

IV. EXPERIMENTS

In the following, we firstly evaluate the performance of the critical point candidates detection method. Then, we evaluate the performance of the proposed SPE-Net (SPE + multi-stream CNN), by training and testing it on the 3D neuron image stacks from a whole mouse brain and the publicly available BigNeuron dataset [34]. Subsequently, an ablation study is conducted to illustrate the contributions of different modules in the multi-stream CNN and the proposed SPE method. Finally, we demonstrate the potential of our detected critical points to be good seed points for neuron reconstruction, by inputting them to an existing seed based neuron reconstruction method.

A. Datasets and Performance Measures

Whole Mouse Brain Sub-image (WMBS): This dataset contains 34 neuron images selected from a whole mouse brain which is provided by the Allen Institute for Brain Science. The

size of these images is $1024 \times 1024 \times 100$ voxels (resolution: $0.2 \times 0.2 \times 1.0 \mu\text{m}/\text{voxel}$). Since the resolution in z dimension of these images is 5 times lower than that in x and y dimensions, we expand the size of the z -axis from 100 to 500 (using cubic interpolation) before processing the images to balance the voxel aspect ratio. We used 12 images for training (WMBS1-12) and 22 images for testing (WMBS13-34).

BigNeuron Project [34]: This is a publicly available dataset for single neuron reconstruction in 3D microscopy images, which consists of different species including fruit fly and other insects, fish, turtle, chicken, mouse, rat, and humans. The sizes of these image stacks are $1018 \times 503 \times 29$, $1600 \times 1600 \times 37$ voxels and so on. We expanded the size of the z -axis for the test BigNeuron images, according to the voxel size information provided by the BigNeuron project. For some images in the BigNeuron project, the voxel size was unknown, and we doubled the size of the z -axis of these images, reflecting the typically lower resolution in the z dimension [16]. We used 30 images from this dataset, 15 for training and 15 for testing.

Performance Measures: In order to evaluate the detection results, we manually labeled the critical points of all the images used in this work, which we used as the gold standard for evaluation. Based on the gold standard, we calculated three commonly used measures from the experimental results: false positive (FP), false negative (FN) and true positive (TP). Specifically, FP represents the number of falsely detected critical points, FN is the number of true critical points that are not detected, and TP represents the number of true critical points detected. Based on those three measurements, we calculated Precision $P = \text{TP}/(\text{TP}+\text{FP})$ and Recall $R = \text{TP}/(\text{TP}+\text{FN})$, as well as the F1-measure, $\text{F1} = 2\text{RP}/(\text{R}+\text{P})$ [35]. Points within ρ voxels to a ground truth were also regarded as true positives. In the experiments, ρ was chosen according to the local neurite diameter of the gold standard, which was estimated by the method proposed in [3]. Specifically, we set $\rho = d(p_{gt})/2 + \eta$, where $d(p_{gt})$ represents the estimated local neurite diameter of a ground truth p_{gt} and η is a “tolerance term”. We set $\eta = 5$ in this paper, taking into consideration the potential diameter estimation error (1 voxel), and the location shifts of the gold standard caused by both the minimum distance discrepancy visible to human eye (2 voxels) [4] and the subjectivity of different annotators when labeling the critical points (2 voxels).

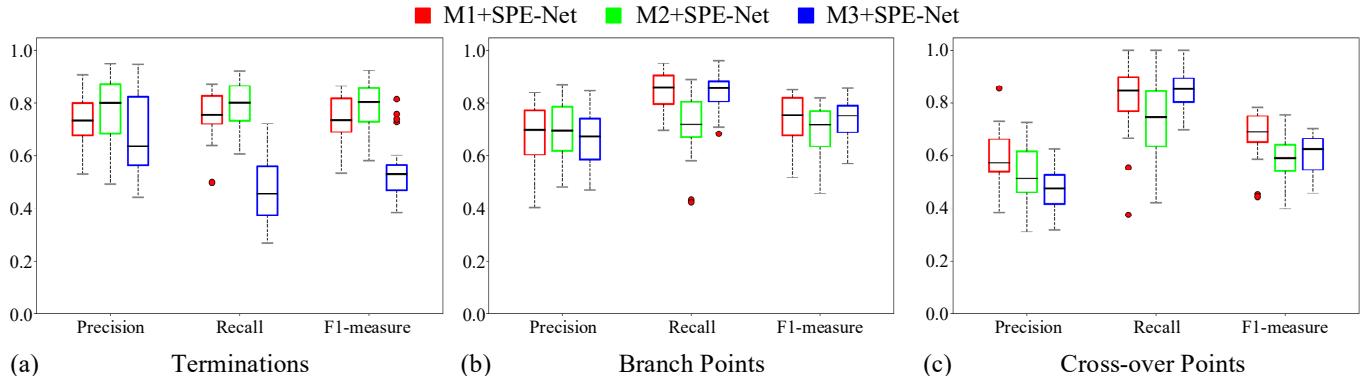


Fig. 7. Performance comparison for different 3D segmentation methods. (a), (b) and (c) are the terminations, branch points and cross-over points detection performances of SPE-Net on the critical point candidates extracted by the compared methods, respectively.

B. Implementation Details

Parameters Selection: The number of spherical surfaces $N = 18$. The range of radii for the spherical surfaces $R = \{3, 4, \dots, 20\}$. The number of azimuth angles and polar angles are 64, i.e., $M_a = M_p = 64$, which means that the input image size of the 2D multi-stream CNN is 64×64 . We set the *expansion ratio* $T = 6$ and *repeating times* $n = 3$ for the inverted residual blocks [36]. The *reduction ratio* for the squeeze-and-excite module is 4 [37]. All the parameters were fixed in the experiments.

Data Augmentation: We rotated the spheres 5 times before extracting the spherical patches at the selected points for training. Specifically, all the spheres generated at the same point were rotated along the x axis by $\pm 90^\circ$ and along the y axis by $\pm 90^\circ$ and 180° . Therefore, 6 training samples can be generated from a point.

Training: The proposed network uses the categorical cross-entropy loss function and Adam optimizer [38]. The initial learning rate is set to $1e^{-3}$, and reduces by 10 times every 4 epochs. All the experiments were carried out on a workstation with an Intel Xeon E5-2683 CPU, 32GB RAM, and a single Titan Xp GPU.

TABLE I
THE AVERAGE RECALL OF THE CRITICAL POINT CANDIDATES
EXTRACTED BY DIFFERENT 3D SEGMENTATION METHODS

Methods	Terminations	Branch Points	Cross-over Points
M1	0.9949	1.0000	1.0000
M2	0.9909	0.9996	0.9999
M3	0.9496	0.9999	0.9999

C. Critical Point Candidates Detection Results

In this section, we evaluate the performance of the critical point candidates detection method, by comparing the improved V-Net (M1) with the Res-Incep-Net [42] (M2) and Attention U-Net [43] (M3). Similar to the improved V-Net, the Res-Incep-Net is also a CNN based 3D neuron image segmentation method. And the Attention U-Net integrates attention gates into the standard U-Net architecture to select the features passed through the skip connections. In the experiments, the compared networks were trained using the same training samples. And the image segmentation results

produced by the compared methods were skeletonized to obtain the critical point candidates set.

Firstly, we compare the recall of M1-M3 to evaluate the proportion of true critical points that are correctly included by the candidate points. From Table I, we can see that all the methods perform well in extracting branch points and cross-over points. But the accuracy of M3 for terminations drops substantially. Secondly, we evaluate the critical points detection results of our SPE-Net on the critical point candidates extracted by M1-M3. From Fig. 7(a), we can observe that M1+SPE-Net and M2+SPE-Net yield close results for terminations, but the recall and F1-measure of M3+SPE-Net drop substantially. This obviously reflects that the relatively low recall rate of M3 for termination candidates affects the subsequent classification. Further, it can be seen from Fig. 7(b) that the three methods show comparable F1-measures for branch points, and M1+SPE-Net performs slightly better. Similar to branch points, there is no significant difference between the three methods for cross-over points, and M1+SPE-Net outperforms the other methods.

To conclude, M1 achieves the overall best performance compared with M2 and M3. And there are close performances between M1+SPE-Net and M2+SPE-Net. The experimental results show that the quality of the critical point candidates have an impact to the critical point classification performance of SPE-Net. But as long as the critical point candidates detection methods can achieve high recall rates, the performance of SPE-Net will be stable. This demonstrates the robustness of SPE-Net to the critical point candidates.

D. Quantitative Results on the WMBS Dataset

In this section, we compare the performance of SPE-Net with the multiscale ray-shooting model (MR) [3] and the DeepBranch (DB) [7], which are two state-of-the-art methods designed for neuron terminations detection and neuron branch points detection, respectively. The parameters for MR were set and tuned according to the authors' suggestion in [3]. Since DB is also a deep-learning based method, we trained it on the same training samples as SPE-Net. In total 4166 training samples were selected from the training images, including 346 terminations, 311 branch points, 154 cross-over points and 3355 non-critical points. The ratio between critical points and non-critical points is imbalanced, because we found that CNNs were "sensitive" to critical points. In other words, a small number of critical point training samples can lead to acceptable recall rates. However, much more non-critical point training samples are required to reduce false positive rates and achieve higher precision rates. One possible explanation for this is that it is easier for the multi-stream CNN to learn the obvious patterns of critical points from the spherical patches. But the patterns of non-critical points are more variable because of the complicated and rich hierarchy of the neuronal structures. As MR and DB were designed for only 1 type of critical points detection, we compared their performances separately. That is, we compared the termination detection results of SPE-Net with MR and the branch points detection results of SPE-Net with DB.

Before quantitative evaluation, we firstly illustrate the 3D critical points detection results of the compared methods on a

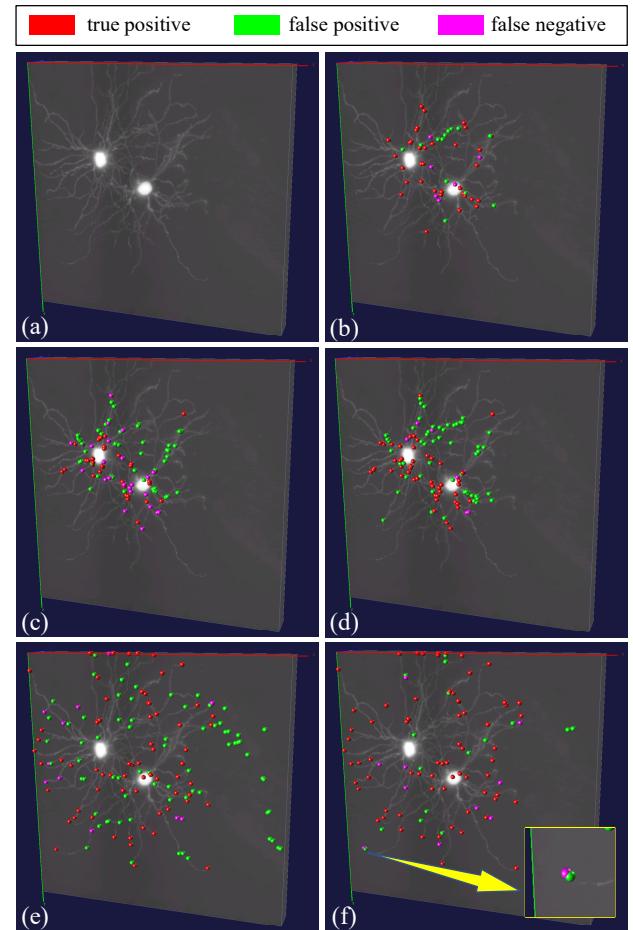


Fig. 8. 3D critical points detection results for a challenging multi-neuron image in the WMBS dataset. The contrast of the images is adjusted slightly for better visualization of weak signals. (a) Original image. (b) Cross-over points detection results by SPE. (c) and (d) Branch points detection results by DB and SPE, respectively. (e) and (f) Terminations detection results by MR and SPE, respectively. Yellow box: the zoomed-in view.

challenging neuron image stack for visual inspection (Fig. 8). It can be seen that two neurons in the image are close to each other, resulting in a high degree of mutual occlusion of the neurite fibers (Fig. 8(a)). Therefore, it is a challenging task to even label the critical points manually. In addition, the weak foreground signals and fuzzy neuronal structures further increase the difficulty of critical points detection. In Fig. 8(c) and (d), we can see that DB fails to detect most of the branch points, whereas SPE-Net only misses a small number of branch points. As for the detection results of terminations (Fig. 8(e) and (f)), MR detects much more false positives than SPE-Net. From the zoomed-in view (yellow box) of Fig. 8(f), we can see that a false positive point is close to a false negative point. This is because SPE-Net detects a termination too far away from the true termination.

Then, we show the quantitative comparison results on the 22 test images from the WMBS dataset in Fig. 9 with box plots³. The detection results of SPE-Net for cross-over points are shown without comparison, because we were not aware of other methods designed for cross-over points detection in neuron image stacks. It can be seen from Fig. 9(a) that our SPE-Net yields higher precision, recall and F1-measure than

³The accuracies of each test images can be found in Supplementary Material.

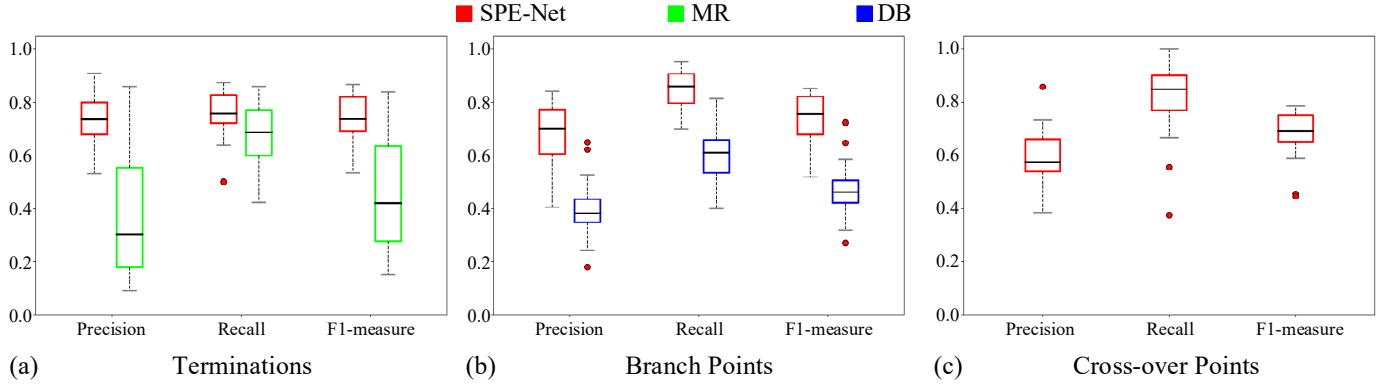


Fig. 9. Performance comparison for the WMBS dataset. (a) Terminations detection performance of SPE-Net compared to MR. (b) Branch points detection performance of SPE-Net compared to DB. (c) Cross-over points detection performance of SPE-Net.

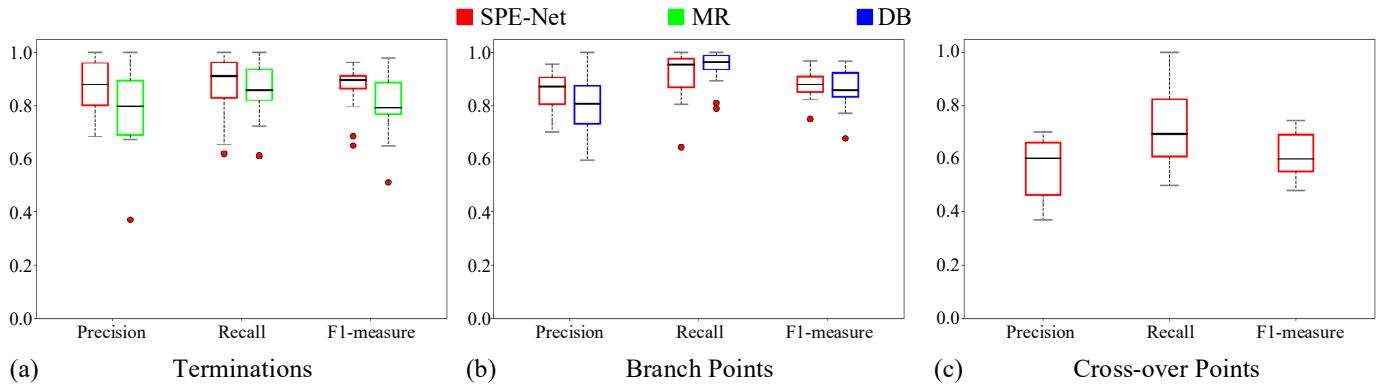


Fig. 10. Performance comparison for the BigNeuron dataset. (a) Terminations detection performance of SPE-Net compared to MR. (b) Branch points detection performance of SPE-Net compared to DB. (c) Cross-over points detection performance of SPE-Net.

TABLE II
AVERAGE ACCURACIES OF DIFFERENT METHODS FOR TERMINATIONS AND BRANCH POINTS DETECTION ON THE WMBS AND BIGNEURON DATASETS

Performance Measures	Datasets	Terminations		Branch Points		Cross-over Points
		SPE-Net	MR	SPE-Net	DB	
Precision	WMBS	0.7190	0.4056	0.6588	0.3808	0.5977
	BigNeuron	0.8706	0.7792	0.8548	0.8066	0.5612
Recall	WMBS	0.7593	0.6801	0.8426	0.6052	0.8182
	BigNeuron	0.8681	0.8553	0.9145	0.9439	0.7191
F1-Measure	WMBS	0.7386	0.5081	0.7395	0.4675	0.6908
	BigNeuron	0.8693	0.8155	0.8836	0.8699	0.6141

MR, which confirms the effectiveness of the proposed method. Note that SPE-Net can detect critical points in the original images, whereas MR requires binarized images as input. Hence, SPE-Net not only performs better, but also has a wider scope of application than MR. It can be seen from Fig. 9(b) that SPE-Net significantly outperforms DB in precision, recall and F1-measure. The experimental results confirm that SPE-Net, which is designed for detecting all 3 types of critical points, outperforms the compared state-of-the-art methods designed for only 1 type of critical points detection.

E. Quantitative Results on the BigNeuron Dataset

After evaluating the performance of SPE-Net on the WMBS dataset, we tested it on the images from the BigNeuron dataset

to further demonstrate the applicability and robustness of the proposed method to different data. 15 images were selected for training and the 15 images reported in the paper of DeepBranch [7] were used as the test images. In Fig. 10, we show the quantitative results on the 15 test images. It can be seen that SPE-Net still outperforms MR in precision, recall and F1-measure. As for branch points detection, SPE-Net achieves comparable results to DB—it yields comparable precision to DB and outperforms DB in recall and F1-measure.

In Table II, we summarize the average precisions, recalls and F1-measures of the compared methods on both WBMS and BigNeuron datasets for a more comprehensive comparison. From the termination detection results, we can clearly see that SPE-Net constantly yields superior results, compared with MR, on both datasets. Also, for branch points

TABLE III
ABLATION STUDY OF THE PROPOSED METHOD

Models	Multi-Stream	Ablation Type					Point Type	F1-Measure
		HS	SE	SPE	TC2.5D	MSMV		
Model 1				✓			Terminations	0.5403
							Branch Points	0.3985
							Cross-over Points	0.4471
Model 2	✓			✓			Terminations	0.6189
							Branch Points	0.5674
							Cross-over Points	0.4846
Model 3	✓	✓		✓			Terminations	0.7056
							Branch Points	0.7281
							Cross-over Points	0.6398
Model 4	✓	✓	✓	✓			Terminations	0.7386
							Branch Points	0.7395
							Cross-over Points	0.6908
Model 5	✓	✓	✓		✓		Terminations	0.4181
							Branch Points	0.5369
							Cross-over Points	0.4094
Model 6	✓	✓	✓		✓		Terminations	0.5504
							Branch Points	0.7214
							Cross-over Points	0.4601

detection, SPE-Net outperforms DB on both datasets, except for the small recall difference on the BigNeuron dataset. The comparison results in Table II demonstrate that SPE-Net performs better than its competitors on both datasets. Further, we can observe that the performances of MR and DB show large varieties on different datasets. Specifically, the F1-measure of MR on the WMBS dataset is about 31% lower than that on the BigNeuron dataset, whereas the figure for SPE-Net is only about 13%. Moreover, the F1-measure of DB drops more dramatically by over 40%. Nevertheless, the figure for SPE-Net is only about 14%, indicating that SPE-Net is more robust than MR and DB on the challenging WMBS dataset. The experimental results show that SPE-Net obtains the overall best performance than the compared methods on different datasets and demonstrate its robustness and wide applicability.

F. Ablation Study

In this section, we conducted an ablation study on the WMBS dataset, as shown in Table III, to illustrate contributions of different modules in the proposed SPE-Net (Model 4). From Model 1 to Model 4, we demonstrate the effectiveness of the multi-stream setting, the h-swish (HS) non-linearity and the squeeze-and-excite (SE) module. From Model 4 to Model 6, the contribution of the proposed SPE is confirmed.

In order to evaluate the contribution of the multi-stream setting, we built a single-stream CNN (Model 1) by using a single *Conv Block A* to accept all the patches. We stacked all the patches together and fed them to the single-stream CNN. Therefore, the input size of the single-stream network is $64 \times 64 \times 18$, in which the third dimension of the input is considered as ‘channel’. By comparing Model 1 and Model 2, we can see that the detection accuracies are improved, which demonstrates the effectiveness of the multi-stream setting. By comparing Model 2 with Model 3, we can see significant

improvements in Model 3, which demonstrates the effectiveness of the HS non-linearity. After adding the SE module, Model 4 achieves the highest F1-measures for all 3 kinds of points.

After verifying the effectiveness of the modules in the multi-stream CNN, we evaluated the contribution of SPE by comparing it with other patches extraction methods. The compared methods are the multi-scale multi-view (MSMV) sampling and the triple-crossing 2.5D (TC2.5D) approach [40]. MSMV is the patches extraction method used in DeepBranch [7], and TC2.5D is an improved version of the 2.5D patches extraction method [41]. In the following, the implemental details of the compared methods are introduced before we discuss the experimental results.

By using Maximum Intensity Projection (MIP), MSMV generates 3 views (axial, sagittal, coronal) of the voxel of interest at 3 different scales (e.g., 20, 30 and 40 voxels), respectively. Therefore, in the experiments, in total 9 slices of size 40×40 were extracted at the voxel of interest. TC2.5D also extracts 9 slices of size 40×40 , including 3D slices from 3 orthogonal directions and 6 diagonal directions. The patch size of the two compared methods was set to 40 to ensure they cover the same area as SPE which covers a spherical region of diameter 40. In the experiments, the patches extracted by MSMV and TC2.5D were fed into our multi-stream CNN. Since the patch size of MSMV and TC2.5D (40×40) is smaller than that of SPE (64×64), we changed the strides from 2 to 1 for the second inverted residual block of *Conv Block B* in the multi-stream CNN. This ensures that the *Output Module* of the multi-stream CNN receives large enough feature maps. The training samples extracted by MSMV and TC2.5D were augmented using scaling and rotation, generating 18 training samples at a point.

From Model 4 to Model 6, we can see that SPE outperforms its competitors by large margins. This is because SPE is able to transform 3D information to 2D without losing 3D structural and contextual information, whereas neither TC-

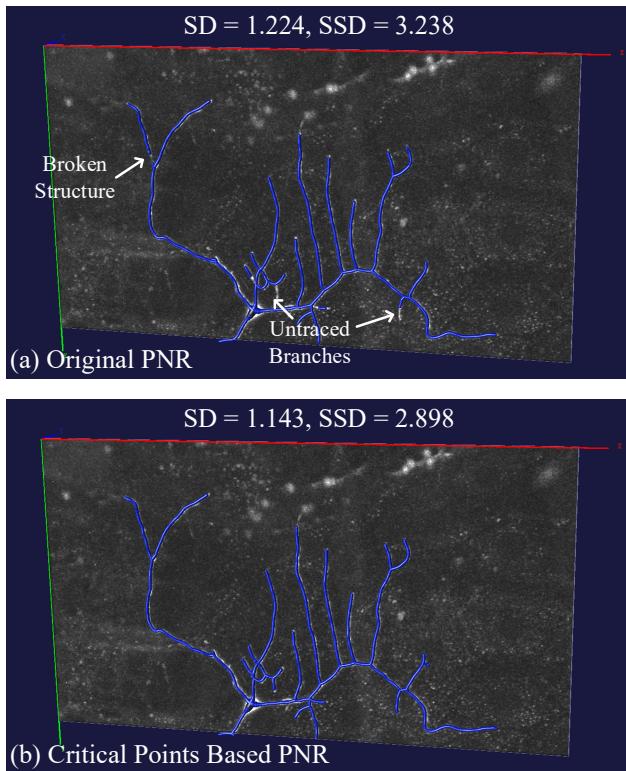


Fig. 11. Visual inspection of neuron reconstruction results produced by PNR, using different seed points. (a) Neuron reconstruction results of the original PNR. (b) Neuron reconstruction results of the critical points based PNR. The threshold S for computing SSD is 2.

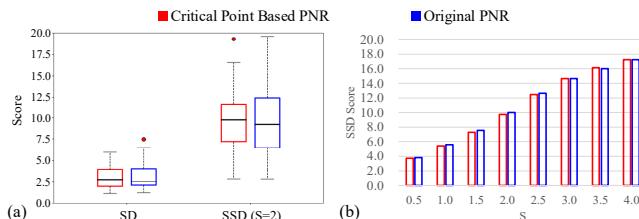


Fig. 12. Neuron reconstruction performance comparison for the test BigNeuron images. (a) The SD and SSD ($S=2$) scores. (b) The average SSD scores calculated using different S.

2.5D nor MSMV can preserve all the 3D information, leading to worse performance in neuron critical points detection.

The ablation study confirms that all the settings of our SPE-Net are effective to improve the critical points detection accuracies and demonstrates the contribution of SPE.

G. Neuron Reconstruction Using the Detected Critical Points

The neuron critical points are helpful to determine the topology and faithfulness of the neuron structure reconstructions [8]. Many reconstruction algorithms and tools can be used to reconstruct the neuron morphology based on the critical points, e.g., the Probabilistic Neuron Reconstructor (PNR) [16]. Starting from the given seed points, PNR recursively traces neuron branches by sequential Monte Carlo estimation, using state transition and measurement models designed specifically for the purpose of neuron tracing. Originally, PNR can automatically detect seed points that have a high probability of being centered at neuronal branches. In order to show the potential of the neuron critical points to be good seed points for neuron reconstruction, we used the

detected critical points as seed points for PNR to trace the neuronal branches in microscopic image stacks.

The images in the BigNeuron dataset were used in the experiments, because manually annotated reconstruction results were provided in this dataset for references. In order to evaluate the neuron reconstruction performances, we used the spatial distance (SD) and substantial spatial distance (SSD) as performance measures, which are widely used to evaluate the neuron reconstruction results [39]. SD measures the mean distance between each pair of closest nodes between two neuron reconstructions (i.e., the manual reconstruction and the reconstruction result generated by computational methods). SSD measures the average distance between pairs of closest nodes when they are at least S voxels apart from each other. The threshold S was set to 2 by default, following the settings in many other studies [20][42].

Illustrative examples of the neuron reconstruction results produced by the original PNR and critical points based PNR are shown in Fig. 11. It can be seen that the critical points based PNR is able to trace more complete neuronal structures than the original PNR, leading to lower SD and SSD scores. In Fig. 12, we show the quantitative comparison of the original PNR and the critical points based PNR on the 15 BigNeuron test images used in Section IV-E. Of these two methods, the critical points based PNR shows a smaller performance spread, indicating that our detected critical points are able to increase the robustness of PNR. Further, by varying the threshold S (Fig. 12(b)), we can observe that the critical points based PNR performs better in most of the cases, which confirms the effectiveness of our detected critical points to PNR.

Indeed, only the position information of our detected critical points was used in the experiments. In other words, the neuron reconstruction performance can be further improved by taking more information provided by the critical points into consideration. For example, starting from a termination, the neuron reconstruction method only needs to trace one path back to the neuron soma or other critical points. Therefore, we can design a path selection method to prevent the tracing method from stepping into the noisy background or the neighboring neuronal structures. Specifically, from all the path candidates starting from a given termination point, we can calculate the confidence scores [20] based on, for example, local intensity or orientation information and select one path with the highest confidence score. This could be done and may be helpful to increase the robustness of the neuron reconstruction methods, but requires further exploration.

V. CONCLUSIONS

In this paper, we proposed a novel 3D neuron critical points detection method for neuron microscopy images. The core idea is extracting and projecting 3D intensity distribution features to 2D by the proposed spherical-patches extraction (SPE) method. A 2D multi-stream CNN is then designed to learn the projected intensity distribution features extracted by SPE and classify any critical point candidate into one of four classes: termination point, branch point, cross-over point, or non-critical point. Experimental results confirm that the proposed method outperforms other state-of-the-art critical points detection methods, and also proved the potential of the

detected neuron critical points to be good seed points for neuron reconstruction.

Additionally, we have established a dataset called Whole Mouse Brain Sub-image (WMBS), in which all the critical points of the images are manually labeled.

In the future, we will add and label more images to enlarge the WMBS dataset. Furthermore, by using this dataset, we are planning to develop algorithms for neuron image analysis such as automatic neuron reconstruction algorithms based on critical points detection and automatic cross-over separation algorithms.

ACKNOWLEDGMENT

The authors would like to thank Dr. Hanchuan Peng from the Allen Institute for Brain Science and Southeast University–Allen Institute Joint Center for sharing the whole mouse brain imaging data.

REFERENCES

- [1] J. De *et al.*, “A graph-theoretical approach for tracing filamentary structures in neuronal and retinal images,” *IEEE Trans. Med. Imag.*, vol. 35, no. 1, pp. 257–272, Jan. 2016.
- [2] E. Meijering, “Neuron tracing in perspective,” *Cytometry A*, vol. 77, no. 7, pp. 693–704, 2010.
- [3] M. Liu, W. Chen, C. Wang, and H. Peng, “A multiscale ray-shooting model for termination detection of tree-like structures in biomedical images,” *IEEE Trans. Med. Imag.*, vol. 38, no. 8, pp. 1923–1934, Aug. 2019.
- [4] H. Peng, F. Long, and G. Myers, “Automatic 3D neuron tracing using all-path pruning,” *Bioinformatics*, vol. 27, no. 13, pp. i239–i247, 2011.
- [5] H. Peng, Z. Zhou, E. Meijering, T. Zhao, G. A. Ascoli, and M. Hawrylycz, “Automatic tracing of ultra-volumes of neuronal images,” *Nature Methods*, vol. 14, no. 4, pp. 332–333, 2017.
- [6] M. Radojević and E. Meijering, “Automated neuron tracing using probability hypothesis density filtering,” *Bioinformatics*, vol. 33, no. 7, pp. 1073–1080, 2017.
- [7] Y. Tan, M. Liu, W. Chen, X. Wang, H. Peng, and Y. Wang, “DeepBranch: deep neural networks for branch point detection in biomedical images,” *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1195–1205, Apr. 2020.
- [8] M. Radojević, I. Smal, and E. Meijering, “Fuzzy-logic based detection and characterization of junctions and terminations in fluorescence microscopy images of neurons,” *Neuroinformatics*, vol. 14, no. 2, pp. 201–219, 2016.
- [9] H. Peng, A. Bria, Z. Zhou, G. Iannello, and F. Long, “Extensible visualization and analysis for multidimensional images using Vaa3D,” *Nature Protocols*, vol. 9, pp. 193–208, Jan. 2014.
- [10] L. Ge, H. Liang, J. Yuan, and D. Thalmann, “Robust 3d hand pose estimation in single depth images: From single-view CNN to multi-view CNNs,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3593–3601.
- [11] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas, “Volumetric and multi-view CNNs for object classification on 3D data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 5648–5656.
- [12] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3D shape recognition,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 945–953.
- [13] Y. Wang *et al.*, “TeraVR empowers precise reconstruction of complete 3-D neuronal morphology in the whole brain,” *Nature Communications*, vol. 10, no. 1, pp. 1–9, 2019.
- [14] J. Xie, T. Zhao, T. Lee, E. Myers, and H. Peng, “Anisotropic path searching for automatic neuron reconstruction,” *Med. Image Anal.*, vol. 15, no. 5, pp. 680–689, 2011.
- [15] G. González, E. Türetken, F. Fleuret, and P. Fua, “Delineating trees in noisy 2D images and 3D image-stacks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010, pp. 2799–2806.
- [16] M. Radojević and E. Meijering, “Automated neuron reconstruction from 3D fluorescence microscopy images using sequential Monte Carlo estimation,” *Neuroinformatics*, vol. 17, no. 3, pp. 423–442, 2019.
- [17] S. Basu, W. T. Ooi, and D. Racocianu, “Neurite tracing with object process,” *IEEE Trans. Med. Imag.*, vol. 35, no. 6, pp. 1443–1451, Jun. 2016.
- [18] S. Mukherjee, B. Condron, and S. T. Acton, “Tubularity flow field—A technique for automatic neuron segmentation,” *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 374–389, Jan. 2015.
- [19] S. Liu, D. Zhang, S. Liu, D. Feng, H. Peng, and W. Cai, “Rivulet: 3D neuron morphology tracing with iterative back-tracking,” *Neuroinformatics*, vol. 14, pp. 387–401, 2016.
- [20] S. Liu, D. Zhang, Y. Song, H. Peng, and W. Cai, “Automated 3D neuron tracing with precise branch erasing and confidence controlled back-tracking,” *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2441–2452, Nov. 2018.
- [21] H. Xiao and H. Peng, “APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree,” *Bioinformatics*, vol. 29, no. 11, pp. 1448–1454, 2013.
- [22] Q. Li and Z. Deng, “A surface-based 3-D dendritic spine detection approach from confocal microscopy images,” *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1223–1230, Mar. 2012.
- [23] R. Su, C. Sun, and T. D. Pham, “Junction detection for linear structures based on Hessian, correlation and shape information,” *Pattern Recognit.*, vol. 45, no. 10, pp. 3695–3706, 2012.
- [24] M. Radojević, I. Smal, W. Niessen, and E. Meijering, “Fuzzy logic based detection of neuron bifurcations in microscopy images,” in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, 2014, pp. 1307–1310.
- [25] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Proc. Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
- [26] M. Liu, H. Luo, Y. Tan, X. Wang, and W. Chen, “Improved V-net based image segmentation for 3D neuron reconstruction,” in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, 2018, pp. 443–448.
- [27] M. Kerschnitzki *et al.*, “Architecture of the osteocyte network correlates with bone material quality,” *J. Bone Mineral Res.*, vol. 28, no. 8, pp. 1837–1845, 2013.
- [28] L. Gu, X. Zhang, H. Zhao, H. Li, and L. Cheng, “Segment 2D and 3D filaments by learning structured and contextual features,” *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 596–606, Feb. 2017.
- [29] K. Jerman, F. Pernuš, B. Likar, and Ž. Špiclin, “Enhancement of vascular structures in 3D and 2D angiographic images,” *IEEE Trans. Med. Imag.*, vol. 35, no. 9, pp. 2107–2118, Sep. 2016.
- [30] A. Sironi, E. Türetken, V. Lepetit, and P. Fua, “Multiscale centerline detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1327–1341, Jul. 2016.
- [31] A. Mosinska, M. Kozinski, and P. Fua, “Joint segmentation and path classification of curvilinear structures,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1515–1521, Jun., 2020.
- [32] T. Ni, L. Xie, H. Zheng, E.K. Fishman, and A.L. Yuille, “Elastic boundary projection for 3D medical image segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2109–2118.
- [33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520.
- [34] H. Peng *et al.*, “BigNeuron: Large-scale 3D neuron reconstruction from optical microscopy images,” *Neuron*, vol. 87, no. 2, pp. 252–256, 2015.
- [35] D. M. W. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” *J. Mach. Learn. Technol.*, vol. 2, pp. 37–63, 2011.
- [36] A. Howard *et al.*, “Searching for MobileNetV3,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1314–1324.
- [37] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7132–7141.
- [38] D.P. Kingma, and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. International Conference on Learning Representations (ICLR)*, vol. 5, 2015.
- [39] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, “V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets,” *Nature Biotechnology*, vol. 28, no. 4, pp. 348–353, 2010.

- [40] S. Liu, D. Zhang, Y. Song, H. Peng, and W. Cai, "Triple-crossing 2.5D convolutional neural network for detecting neuronal arbours in 3D microscopic images," in *Proc. Proc. Med. Image Comput. Comput.-Assist. Intervent. Workshop (MICCAIW)*, 2017, pp. 185–193.
- [41] H. R. Roth *et al.*, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1170–1181, May 2016.
- [42] R. Li, T. Zeng, H. Peng, and S. Ji, "Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction," *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1533–1541, Jul. 2017.
- [43] O. Oktay *et al.*, "Attention U-Net: Learning where to look for the pancreas," in *1st Conference on Medical Imaging with Deep Learning*, 2018.