DeepBranch: Deep Neural Networks for Branch Point Detection in Biomedical Images

Yinghui Tan, Min Liu, Weixun Chen, Xueping Wang, Hanchuan Peng, and Yaonan Wang

***Abstract*— Morphology reconstruction of tree-like structures in volumetric images, such as neurons, retinal blood vessels, and bronchi, is of fundamental interest for biomedical research. 3D branch points play an important role in many reconstruction applications, especially for graph-based or seed-based reconstruction methods and can help to visualize the morphology structures. There are a few hand-crafted models proposed to detect the branch points. However, they are highly dependent on the empirical setting of the parameters for different images. In this paper, we propose a DeepBranch model for branch point detection with two-level designed convolutional networks, a candidate region segmenter and a false positive reducer. On the first level, an improved 3D U-Net model with anisotropic convolution kernels is employed to detect initial candidates. Compared with the traditional sliding window strategy, the improved 3D U-Net can avoid massive redundant computations and dramatically speed up the detection process by employing dense-inference with fully convolutional neural networks (FCN). On the second level, a method based on multi-scale multi-view convolutional neural networks (MSMV-Net) is proposed for false positive reduction by feeding multi-scale views of 3D volumes into multiple streams of 2D convolution neural networks (CNNs), which can take full advantage of spatial contextual information as well as fit different sizes. Experiments on multiple 3D biomedical images of neurons, retinal blood vessels and bronchi confirm that the proposed 3D branch point detection method outperforms other state-of-the-art detection methods, and is helpful for graph-based or seed-based reconstruction methods.**

***Index Terms*— Branch points, neuron reconstruction, convolutional neural networks cascade.**

I. Introduction

IN BIOMEDICAL research, morphology reconstruction of tree-like structures such as neurons, retinal blood vessels and bronchi is very important [1], [2]. In neurobiology research, 3D neuron reconstruction on a large dataset of volumetric microscopy images is essential to understand the function of brain's neural networks [1]. In ophthalmology, the morphology of retinal vessel trees can provide important clinical information in diagnosing diseases such as glaucoma, proliferative diabetic retinopathy and so on. What's more, the morphological reconstruction of bronchi is of great significance in the researches of various pulmonary diseases, and it enables quantitative investigation of bronchus pathologies. In order to study the morphology of tree-like structures in biomedical images, digital reconstruction (tracing) is required.

Many studies have so far focused on the digital reconstruction of tree-like structures from biomedical images [2], [5]–[7]. A method for automatically and accurately detecting critical points (terminations and branch points) of tree-like structures is very helpful to many digital reconstruction approaches for tree-like structures [3], [8]–[11]. However, the topology of tree-like structures in volumetric images is usually complicated such as in Fig. 8 (a2), and it is a challenging task to obtain a complete and precise reconstruction of the tree-like structures by using only terminations in most graph theory based reconstruction methods [4]. The difficulty of reconstructing tree-like structures would be greatly reduced if the branch points could be detected automatically. As shown in Fig. 1, those branch points are identified by the user to directly trace the neuron structure and further identify how the neuron branches into finer arbors. Besides, the analysis of branch points is of significance in many applications, such as neuroanatomic studies, cardiovascular diseases researches and landmark identification in image registration [12].

However, human visual recognition of branch points in tree-like structures is time-consuming as the structures are usually complex and the number of branch points is large. Fig. 2 (a)-(c) show examples where the branch points should be confirmed by manually observing a bronchus structure along different direction. Otherwise, a false negative point would be produced because of occlusion from other structures. (d)-(f) show two false positive branch points due to erroneous visual identification of structure intersections, which could be prevented by zooming in and rotating the neuron structures a few times. Therefore, manual labeling of branch points in tree-like structures is very difficult.

It is a challenging task for the regular interest point detection methods [13] to directly detect 3D branch points in biomedical images, and the existing methods for detecting neuron branch point or vessel bifurcation detection [14], [15] often focus on only one particular type of dataset. 3D interest point detection methods such as 3D-Sift [16], 3D-Harris [17], Heat Kernel

Manuscript received July 31, 2019; revised September 13, 2019 and September 30, 2019; accepted October 3, 2019. Date of publication October 7, 2019; date of current version April 1, 2020. *(Corresponding author: Min Liu.)*

Y. Tan, M. Liu, W. Chen, X. Wang, and Y. Wang are with the College of Electrical and Information Engineering, Hunan University, Changsha 410082, China, and also with the National Engineering Laboratory for Robot Visual Perception and Control Technology, Changsha 410082, China (e-mail: liu_min@hnu.edu.cn).

H. Peng is with the Allen Institute for Brain Science, Seattle, WA 98109 USA, and also with Southeast University-Allen Institute Joint Center, Nanjing 210096, China (e-mail: h@braintell.org; hanchuanpeng@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2019.2945980

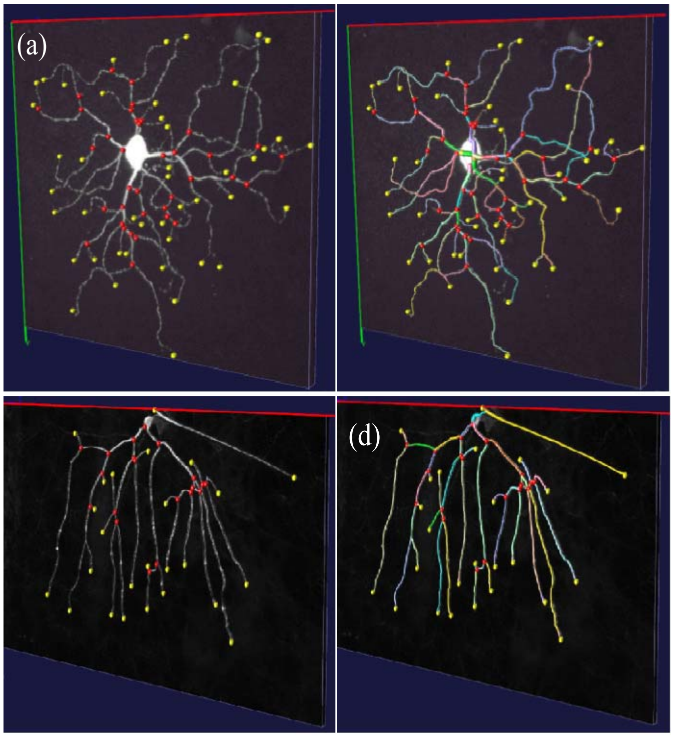

Fig. 1. Vaa3D-Neuron [45] tracing based on 3D critical points. (a) and (c) A mouse retinal ganglion cell image and a fruit fly retinal neuron image with indicated branch points (red spheres) and terminations (yellow spheres). (b) and (d) Reconstructed neuron structures based on the critical points in (a) and (c). (Colored segments are the reconstructed neurite structures.).

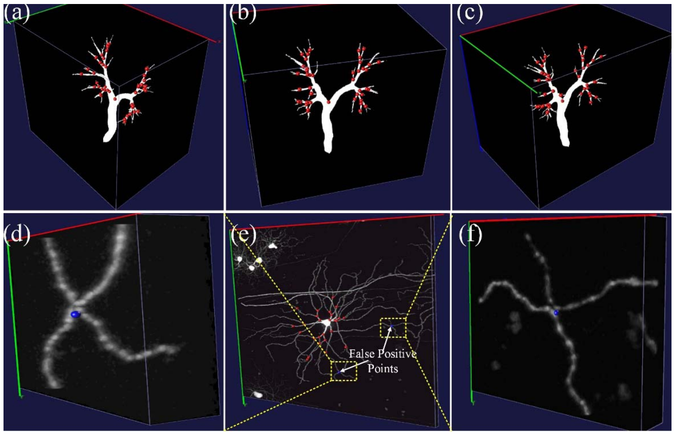

Fig. 2. Visual inspections of manually labeled branch points. (a)-(c) are different views of a bronchus image with complicated spatial structures. (e) is a mouse retinal ganglion cell image. (d) and (f) are two false positive points (blue spheres) in (e).

Signature [18] are not capable of detecting 3D branch points directly. A directional filtering and feature extraction algorithm in combination with a two-stage fuzzy-logic based reasoning system was proposed in [3] to detect the 2D terminations and branch points for neurons. However, it has not been extended to detect 3D branch points. To detect 3D terminations, a ray-shooting intensity distribution model was proposed [19], [20]. However, the performance of such detection methods is very sensitive to the selection of hyper-parameters. It is not an easy task to tune the parameters for different biomedical images containing tree-like structures.

Recently, inspired by a large amount of available data and more powerful computational resources, especially the parallelization ability empowered by Graphic Processing Units (GPUs), CNNs [21] have shown their abilities of outperforming the state-of-the-art methods in classical computer vision applications [22], [23], [25]–[27]. Actually, CNNs have achieved great success with hierarchical feature representations in 2D medical images, such as hemorrhage detection in blood vessel images [28], retinal vessel segmentation [29] and lung disease detection [30].

However, the effective application of CNNs on 3D volumetric data still remains an open problem. One way is to employ 2.5D CNNs [31], which use slices of 2.5D patches as different input channels of 2D CNNs. However, the missing 3D structure information may constrain their performance. Therefore, the multi-view CNNs based methods [32], [33] have attracted much attention in some analysis tasks of 3D volumetric images. In those methods, volumetric images are projected to fixed views (planes), followed by each view being processed by 2D CNNs and finally integrated using multi-view fusion. However, those methods were still limited to 2D kernels and cannot fully utilize spatial information. To overcome this shortcoming, some research applied deep 3D CNNs for both detection and classification of medical imaging data [34], [35]. For example, DeepVesselNet [36] trained 3D CNNs with point labels, which specify a single voxel location that indicates the presence of a vessel bifucation. However, 3D CNNs come with an increased number of parameters and significant memory and computational requirements. Besides, pixel classification using 3D CNNs would result in inefficiency in the detection task due to repeated computations of convolutions on the same voxels in overlapping patches.

To alleviate these problems, this paper proposes a two-level cascaded framework for branch point detection in biomedical images, the diagram of which is shown in Fig. 3. In our framework, candidate regions containing branch points are firstly detected by using segmentation technology [37]. An improved 3D U-Net [38] model designed on the basis of fully convolutional networks is employed in the segmentation task. On the one hand, it can produce dense outputs with the same size as inputs instead of performing pixel classification, which greatly decreases the computation cost. On the other hand, it can extract features that contain rich local and global information by integrating features of different levels. Then, a classification procedure is followed to separate true branch points from false positive points. In the proposed approach, we adopt a deep multi-view learning framework for false positive reduction, in which a multi-scale sampling strategy is employed to accommodate the branch size diversity, and the outputs are combined using a dedicated fusion method to obtain the final branch point detection results.

II. Methods

A two-level designed convolutional network is adopted in our approach for branch point detection, including a candidate

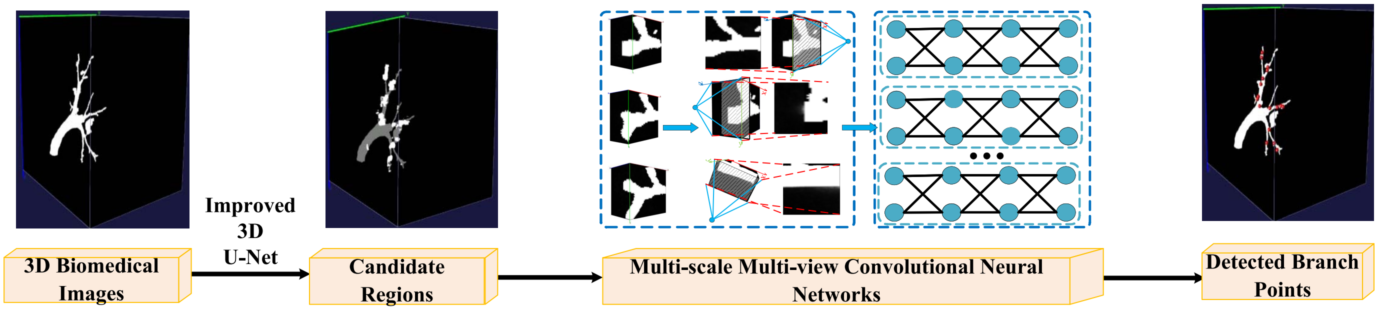

Fig. 3. Diagram of the proposed branch point detection method in biomedical images.

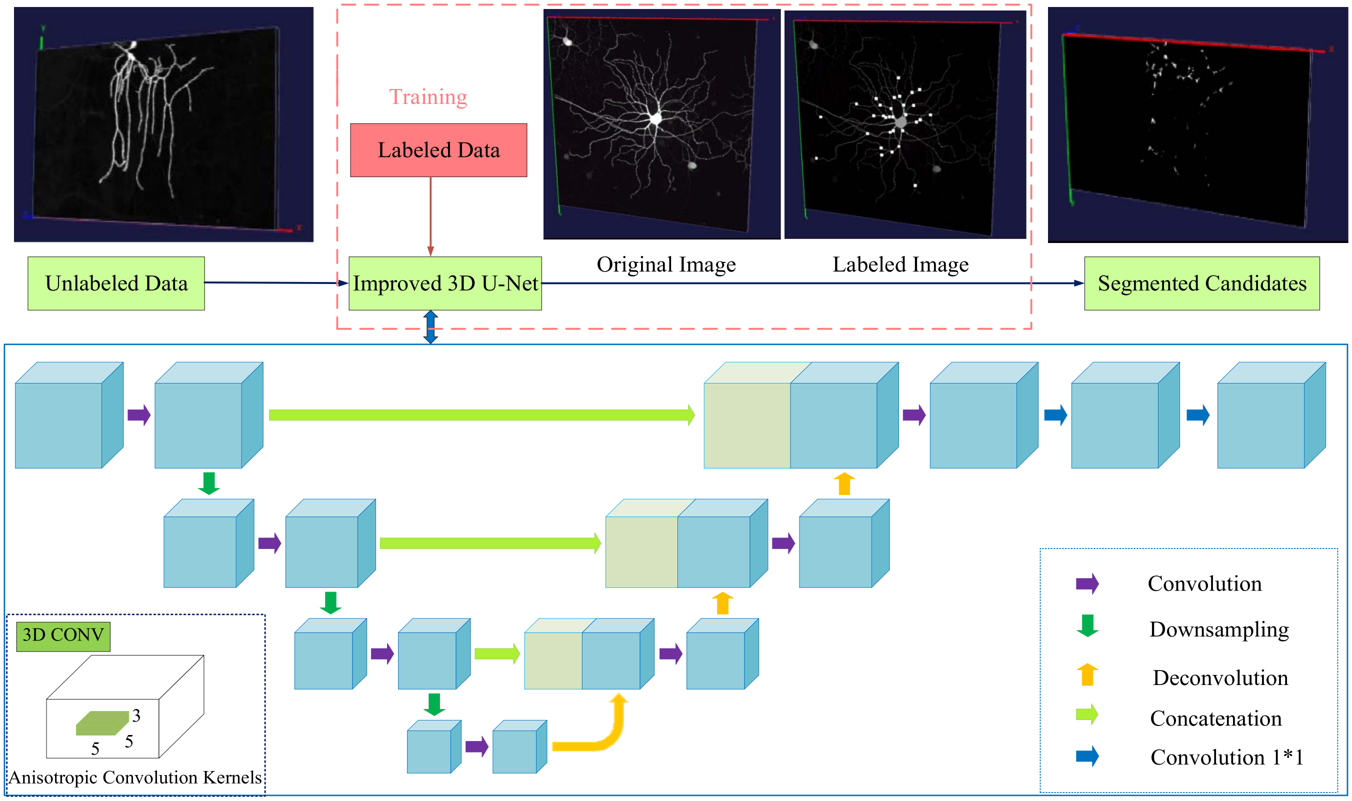

Fig. 4. Detailed architecture for candidate segmentation using improved 3D U-Net. The architecture of the improved 3D U-Net is shown in the blue box, which has three downsampling and upsampling stages. In order to accommodate unbalanced image sizes, the convolution kernels in the original 3D U-Net are replaced by anisotropic convolution kernels. The labeled image is generated by using the method described in Section II-A.

region segmenter and a false positive reducer. The details for each level are expanded in the following subsections.

A. Candidate Region Segmentation Based on 3D U-Net

A suitable candidate segmenter for detection tasks should be fast and highly sensitive in order to detect branch point. Meanwhile, the number of branch point candidates must be within a reasonable limited range. To achieve these objectives, the branch point candidate detection task is formulated as a segmentation problem, in which an improved 3D U-Net [38] model with anisotropic convolution kernels is used as a candidate segmenter, as shown in Fig. 4. U-Net, as a typical representation of FCN, can make good use of information from different levels and can detect multi-scaled branch points from different datasets. Meanwhile, the process for candidate detection is speeded up, as the candidate segmenter can provide dense predictions. Additionally, we can obtain coarse branch detection results by performing connected domain detection on the regional results of the first level.

1) Sampling: The sizes of source image stacks show great variation, for example, the size of a mouse retinal ganglion cell image may be up to $2048 \times 2048 \times 128$ pixels, while the size of a segmented bronchus images may only be $123 \times 126 \times 299$ pixels. Due to the limited GPU resources and highly different image sizes, we cannot use the whole original image as the input of the networks. Besides, there is a serious imbalance between positive and negative samples when using the whole image as the input, as the branch point regions (positive samples) only occupy a small proportion of pixels in the image. Therefore, from the original image, we extract image

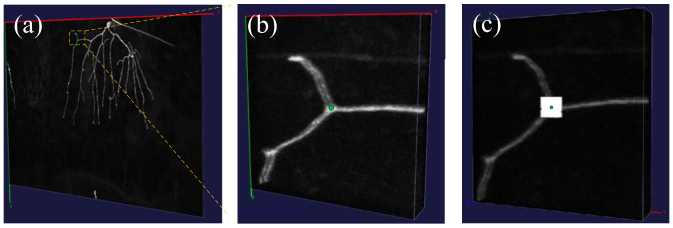

Fig. 5. Example of a branch point and its label. (a) A fruit fly neuron image. (b) A branch point from (a) (the green point). (c) The label of (b) (the bright square centered on the green point).

patches with size of 64 × 64 × 32 pixels as our samples for training the segmentation networks. When extracting a positive sample that includes at least one branch point, we take the coordinates of the marked branch point as the center and obtain a patch with size of 64 × 64 × 32 pixels as a positive sample. A random small coordinate offset is added to increase the sample's diversity. Patches cut off at random coordinates act as negative samples. The positive samples for training are further augmented by scaling and rotation.

Usually, accurate pixel-by-pixel annotation information is required in the training phase of a segmentation networks. However, it is unnecessary to do this in our task for detecting the initial candidate regions containing branch points. We just need to annotate those points which specify a voxel location indicating that there is a branch point. These annotations are then used along with the local diameters (estimated on the basis of experience) to generate the required labels. The size of the labelled image is the same as the size of the original image. And in each labeled image, with the marked branch point position as the center, the value within the radius is set to 1, and the rest is set to 0. Fig. 5 shows an example of a branch point and its label.

2) Networks Architecture: As the images containing tree-like structures come from different datasets (neuron, bronchus, and retinal blood vessels), which are contributed by different research groups and acquired using different imaging techniques, the images show great differences of image quality, size and structure. To segment candidate regions, the complicated structures of biomedical images require the networks to learn features of unbalanced size, since the images have different size in horizontal, vertical and z-axis. To overcome this difficulty, we propose to use improved 3D U-Net based on fully convolutional layers together with deconvolutional layers for candidate segmentation.

As illustrated in Fig. 4, the improved 3D U-Net is composed of a contracting path (left-side) and an expansive path (right-side). As the image resolution of biomedical images may be unbalanced (z resolution is much less than xy resolution), anisotropic convolution kernels (5 × 5 × 3 filters) considers a trade-off among receptive field, inference time and memory efficiency [39]. In the downsampling stage, the pooling layer is replaced by 3 × 3 × 3 convolution kernels with stride of 2 × 2 × 2, which can reduce the size of the input signals as well as increase the receptive field of the computing features in the subsequent networks layer.

In the right part of the improved 3D U-Net, the features extracted from the same stage of the left part are concatenated into the right portion to provide local and global information for candidate segmentation. By this means, the integration of features from different levels can also help to accommodate the branch size diversity. Deconvolution layers are used for up-sampling the feature maps and increasing the size, which is contributing to output candidate segmentation results with the same size as the input. The classification layer is implemented as fully convolutions with 1 × 1 × 1 kernel size, which are capable of dense-inference and can produce outputs of the same size as the input volume. The feature maps computed by this layer are converted to probabilistic segmentations of the candidate regions and non-branch regions.

B. False Positive Reduction Based on MSMV-Net

Each image containing tree-like structures is sent into the segmentation networks, then the output probability map is binarized to obtain the branch point candidates. In the false positive reduction step, we use a multi-scale multi-view convolutional network (MSMV-Net) to classify each voxel of candidate regions as either a branch point or a false positive, the diagram of this process is shown in Fig. 6. In addition, we have also added a multi-scale sampling strategy to accommodate the branch size diversity, as the branch points from different image stacks may vary in size.

1) Sampling: Multi-view CNNs have been proven efficient in 3D object classification [47]. However, in contrast to object classification, point type classification of tree-like structures is related to not only the local intensity distributions, but also to the positions and surrounding anatomical structures. So, for the i_{th} voxel, a 3D block V_i is first extracted to represent the needed information. Here it is important to note that image patches are in 3D, whereas the inputs to the networks are 2D images. In order to combine information across all three views of the block, we firstly compute the Maximum Intensity Projection (MIP) of the 3D block V_i across each view (axial, sagittal, coronal) and form patches consisting of three 2D slices $S_{i1}, S_{i2}, S_{i3}\}$. Considering that the radius of the tree-like structures may vary in size in different images, we employ a multi-scale sampling strategy to ensure that each point and its surroundings are captured. In total, for each manually annotated branch point, we use 3 scales (20 × 20 pixels, 30 × 30 pixels and 40 × 40 pixels), with 3 views for each scale, leading finally to 9 views. All 9 views are then fed into multiple 2D CNN streams with shared weights, while the negative samples are randomly selected from non-branch or noisy areas. We ensure that one third of the training set is real branch point.

2) Networks Architecture: The CNNs used for branch point detection from the candidate regions are constructed by combining various streams of CNNs, referred as a multi-view architecture. Each stream processes patches from a specific view, for which the outputs are combined using feature fusion to obtain the branch point results.

As depicted in Fig. 6, each view is fed into stacks of five convolutional layers. And each stack consists of 96, 128, 128,

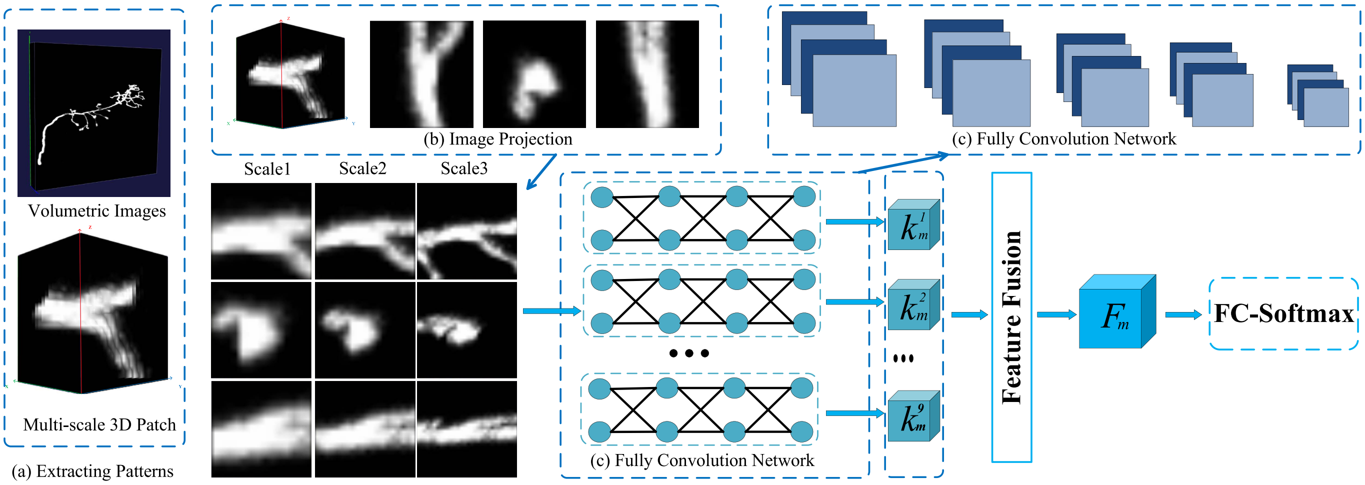

Fig. 6. Diagram of the proposed multi-scale multi-view convolutional neural networks for false positive reduction. (a) Extracting patterns. (b) Image projection of (a). (c) Proposed fully convolutional networks for feature learning.

256 and 128 filters of size $5 \times 5, 5 \times 5, 3 \times 3, 3 \times 3$ and 3×3 respectively. As the pooling layer replaces a region of an image by a statistic (e.g. mean, or maximum) of that region, and also leads to information losses of fine local details which are very crucial in pixel-wise detection tasks of medical images, we removed all pooling layers to perform this pixel-wise learning task. Each convolution layer is followed by a Rectified Linear Unit (ReLU) layer to avoid the vanishing gradient problem.

To avoid the training from over-fitting, a 50% Gaussian Dropout rate on all fully connected layers as well as $L2$ regularization with λ_2= 0.0001 are applied. To train the networks, we use the stochastic gradient descent algorithm with the Adam update rule as the stochastic optimization strategy, a batch size of 1000 and a categorical cross-entropy loss function. We use a decaying learning rate starting from $1e^{-4}$ and gradually decreased to $1e^{-7}$ on the last epoch. We use an early stopping policy by monitoring validation performance and picked the best model with the highest accuracy on the validation set.

3) Feature Fusion: Specifically, in the feature fusion layer, the appearance features produced by multiple 2D CNNs streams $\left\{K_m^{(r)} \in \mathbb{R}^{I_1 \times I_2 \times I_3} | r = 1, 2, \ldots, 9\right\}$ are aggregated to obtain an activation feature F_m representing the whole views, as shown in Fig. 6. $I_1 I_2$ and I_3 are the size and number of channels of each appearance feature produced by each 2D CNN, respectively. The element-wise maximum operation is used across the multiple-view projection images to select the maximum activation of each element of the multi-view appearance feature maps in the feature fusion layers, as shown in Equation 1:

$$F_{m,c} = max(K_{m,c}^1, K_{m,c}^2, \ldots\ldots, K_{m,c}^9) \tag{1}$$

where c is the c_{th} channel of the fused activation feature F_m, recorded as $F_{m,c}$. The feature fusion F_m is then sent to the remaining fully connected layer and softmax layer, which are used to determine the existence of branch points. In this step, we select 0.5 as the threshold probability to separate branch points from false positive points. In this way, the regional results of the first level of the two-level framework are converted into coordinate results.

III. Experiment

To validate the performance of the proposed method for branch point detection, we trained and tested the proposed framework on 3D neuron image stacks from the BigNeuron Project [42], as well as 3D bronchus images and retinal blood vessel images [1] after we annotated the branch point ground truths manually. Rotating and scaling was conducted on 1770 branch points to form 28320 positive samples for training. We firstly evaluate the performance of the improved 3D U-net for candidate segmentation, and obtain coarse results based on the segmentation results by using connected domain algorithms. Then, we present and discuss the final branch point detection results on neuron image stacks, bronchus images and retinal blood vessel images respectively by sending the segmented results to the MSMV-Net. Finally, we conduct experiments to explore the application of branch points on reconstruction of tree-like structures. All mentioned neural networks architectures were implemented with Python under the framework of TensorFlow [40]. The hardware configurations were: a NVIDIA GeForce GTX 1070TI GPU, an Intel Xeon E5-2683 CPU and 16 GB RAM. The source code and models of our method are available from https://github.com/tyhhnu/DeepBranch.

A. Performance Measures

In order to demonstrate the detection performance of the proposed branch point method, three commonly used measurements are computed in the experiments: true positive (TP), false positive (FP) and false negative (FN). Specifically, TP represents the number of true branch points detected, FP represents the number of falsely detected branch points and FN is the number of true branch points that are not detected. Based on those three measurements, we calculate *Precision* = TP/(TP+FP) and *Recall* = TP/(TP+FN), which are used to

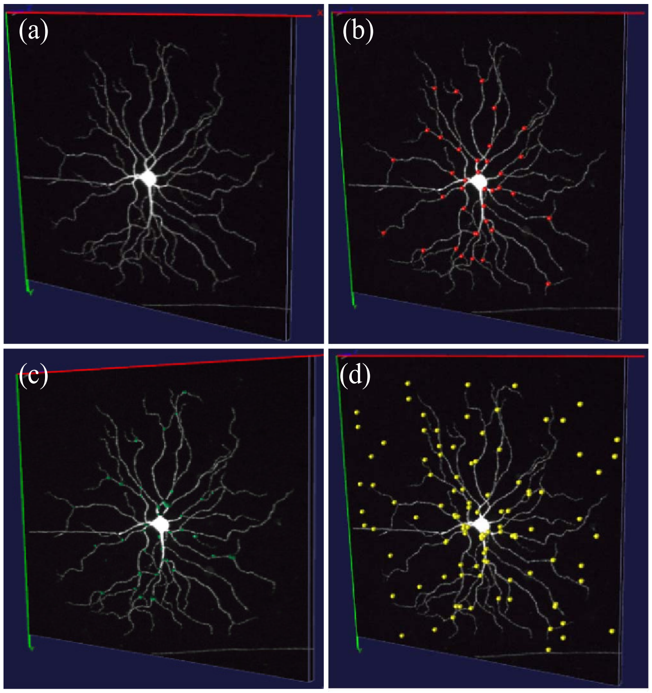

Fig. 7. (a) A 3D mouse neuron image stack. (b) Ground truth of branch points (red spheres). (c) Segmented candidates of (a) (green surface). (d) Coarse result of branch point using connected domain detection on (c) (yellow spheres).

evaluate the detection performance of different approaches. Points that are within ε pixels of a ground truth are also regarded as true positives. In the experiments, ε is empirically set to be 5 or 10 pixels for images with different size.

B. Candidate Segmentation Results

As mentioned in section II-A, to train the candidate segmentation networks, each image containing tree-like structures is cut into 3D blocks with size of $64 \times 64 \times 32$ pixels. We ensured that each 3D block contains at least one manually labeled branch point. But in the testing phase, the whole testing image of any size is fed into the improved 3D U-Net to produce the probability maps within a few seconds. The results of the segmentation networks are then binarized with a probability threshold of 0.5. The results of candidate segmentation are depicted in Fig. 7 (c), where we can see that the candidate regions covered almost all of the branch points. We can obtain coarse detection results by performing connected domain detection on the segmentation results, in which we use the geometric center point of each isolate connected domain as the branch point, as depicted in Fig. 7 (d). We can see that there are still lots of false positive points, which need further verification.

In order to evaluate the results from the first level of the proposed networks, 100 biomedical images containing tree-like structures (including 40 3D neuron images, 30 bronchus images and 30 blood vessel images) are selected as the test set. The sensitivity of the candidate segmenter and precision/recall rates of the coarse results are shown in Table I. Sensitivity is calculated as $Sensitivity = \mathrm{N_c} / \mathrm{GT}$, in which $\mathrm{N_c}$ is true positive point covered by the candidate segmentation results, and GT is the ground truth. We can see that the candidate segmenter yields a significantly high sensitivity rate, which demonstrates that most of the branch points are extracted by the segmentation stage. The average recall rate is low, because the geometric center point may deviate from the true branch point to form false positive points, as no structure information is used in the process of generating the coarse detection results. And in neuron images, the average precision rate of the coarse detection result is relatively low, because the neuron images contain much more noise and dense, complicated structures than the other images. Therefore, it is necessary to add a false positive reduction step to detect the true branch points from the candidate segmentation results, which would be completed within about a minute.

TABLE I
SENSITIVITY OF CANDIDATE SEGEMENTER AND AVERAGE PRECISION/RECALL RATES OF THE COARSE RESULTS

Image Stacks	Number of Images	Average Precision (%)	Average Recall (%)	Candidate Segmenter Sensitivity(%)
Neuron	40	69.61	76.73	89.91
Bronchus	30	81.31	82.59	97.52
Vessel	30	91.10	90.69	97.73
Average		79.57	80.67	95.05

C. Evaluation on 3D Neuron Images

We compared the detection results of our proposed method with other detection methods, including deep learning methods and traditional methods. On the one hand, the method based on 2.5D CNNs was used for comparative experiments. In this method, slices of 2.5D patches are fed into CNNs as different input channels. On the other hand, we also compared our CNNs cascade method with All-Path-Pruning (APP2) [41]. APP2, as an automatic neuron reconstruction algorithm, was not designed for branch point detection. However, the branch points can be easily extracted from the APP2 reconstruction outputs by identifying the nodes with more than one parent or children from the reconstruction results (which is actually a tree representing the neuron morphology). This evaluation method was also used in [3].

The 3D neuron images in this study were selected from the gold166 dataset in BigNeuron Project [42], which consists of different species including fruit fly and other insects, fish, turtle, chicken, mouse, rat, and humans. Those images contain expert manual neuron reconstruction results for reference, so they are widely used for testing. Those selected testing data contains neuron image stacks of different sizes, for example $511 \times 511 \times 505$ pixels, $724 \times 1024 \times 33$ pixels, $1018 \times 543 \times 29$ pixels, $2047 \times 890 \times 23$ pixels and so on.

Some typical results of the detected 3D branch points on the Bigneuron5 neuron stack and Bigneuron7 stack are shown in Fig. 8, where (b1)-(d1) and (b2)-(d2) show the detection results obtained using different methods, such as the CNNs cascade method (b1)(b2), the 2.5D CNNs based method (c1)(c2) and APP2 (d1)(d2). The branch point

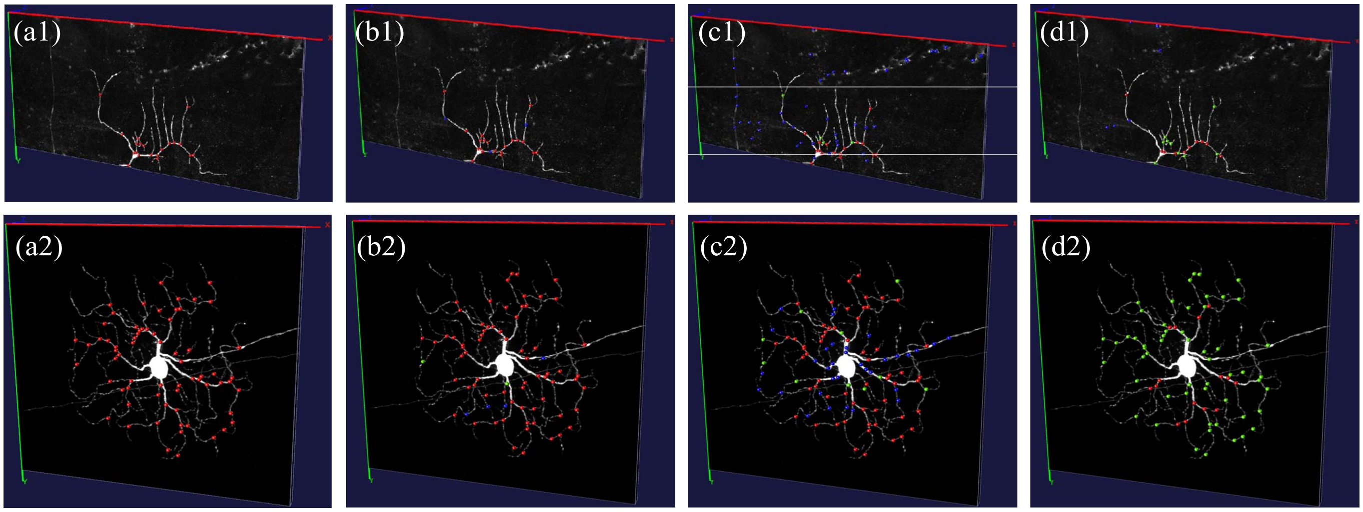

Fig. 8. Examples of 3D branch point detection results in fruit fly and mouse neuron images. The red, blue, and green spheres indicate true positive points, false positive points, and false negative points, respectively. (a1) and (a2) Ground truth of branch points in neuron images. (b1) and (b2) Branch point detection results using the proposed method. (c1) and (c2) Branch point detection results using 2.5D CNN. (d1) and (d2) Branch detection point results using APP2.

detection in those two images is very difficult even for human experts, as they contain discontinued neurite structures and strong background noise around the neurite structures. From Fig. 8, we can see that the CNNs cascade method is able to detect most of the 3D neuron branch points and reject false positive points effectively, especially in challenging image stacks which contain massive noise and complicated neuron structures. The result obtained using the 2.5D CNNs based method contains too much false positive points, coming from false positive 2D branch points caused by noise, rough surfaces and overlapping in the MIP images. The CNNs cascade based method (M1) was tested on a total of 40 3D neuron images, and a comparison of the detection results of 2.5D CNNs (M2) and APP2 (M3), for 15 typical neuron images is shown in Table II.

Note that the recall rate of the proposed method (M1) outperforms slightly than 2.5D CNNs based method (M2), but is much better than APP2 (M3). Noise is one of the main reasons accounting for APP2's poor performance in neuron images. Besides, APP2 also missed some branches of weak signals, resulting from the downsampling and pruning process of APP2, but it is easy for the 2.5D method to detect branch points in MIP images. From the comparison of false positive rates and false negative rates of 15 neuron images in Table II, we can see that our proposed method outperforms the 2.5D CNNs based method and APP2. Especially for image BigNeuron3, the proposed method (M1) gains a detection accuracy rate by at least 10% over the other methods.

D. Evaluation on 3D Bronchus Images

In this section, we show the application of our method in 3D bronchus images. The testing images were segmented from lung CT images obtained from hospital [43]. The sizes of the segmented bronchus images varied from 123×126×299 pixels to 326×234×576 pixels. The detection results of the proposed method for two segmented 3D bronchus images are shown in Fig. 9, where it can be seen that most of the branch points are correctly detected. A comparison of the detection results of 2.5D CNNs (M2), APP2 (M3) and the proposed method for 9 bronchus images is shown in Table III. We can see that the CNNs cascade based method (M1) also outperforms M2 and M3 in the application of bronchus images.

From Table III, we can see that the precision rates of APP2 in bronchus images is not as good as in the neuron images. That is because during the process of reconstructing the morphology structures of bronchus images, APP2 tends to 'branch' at the thick trachea area. This would result in a lot of false positive points when identifying branch points from the reconstruction result. The experimental results on

TABLE II
PRECISION AND RECALL RATES OF DIFFERENT METHODS FOR 3D NEURON IMAGE STACKS

Image Stacks	Precision (%)			Recall (%)		
	M1	M2	M3	M1	M2	M3
Bigneuron1	90.00	65.85	79.41	96.43	90.00	72.97
Bigneuron2	70.59	44.44	85.71	100.00	75.00	70.59
Bigneuron3	100.00	53.57	83.33	93.75	83.33	71.43
Bigneuron4	73.91	42.50	85.00	80.95	73.91	58.62
Bigneuron5	86.96	40.00	80.00	100.00	86.96	66.67
Bigneuron6	94.74	64.29	94.74	97.30	87.80	57.60
Bigneuron7	72.22	61.90	89.66	100.00	78.79	66.67
Bigneuron8	75.00	62.26	91.67	100.00	89.19	57.89
Bigneuron9	84.62	65.67	86.27	97.78	88.00	61.97
Bigneuron10	59.42	65.08	87.23	78.85	85.42	63.08
Bigneuron11	72.22	67.24	82.98	97.50	84.78	61.90
Bigneuron12	80.65	62.50	75.76	89.29	80.65	59.52
Bigneuron13	87.50	56.76	72.41	95.45	77.78	61.76
Bigneuron14	75.00	60.00	67.74	95.45	75.00	61.76
Bigneuron15	87.10	67.50	77.14	93.10	84.38	69.23
Average	79.57	60.03	83.81	94.02	83.97	62.59

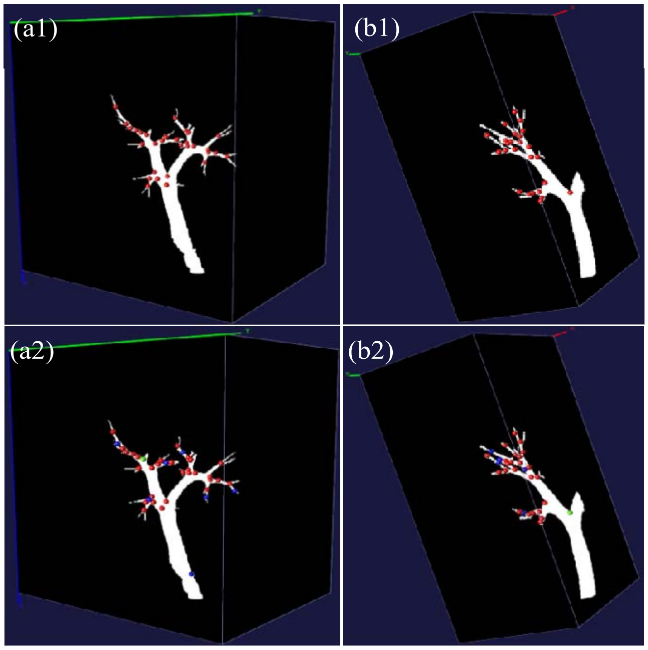

Fig. 9. Examples of 3D branch point detection results in bronchus images using the proposed method. The red, blue, and green spheres indicate true positives, false positives, and false negatives, respectively. (a1) and (b1) Ground truth of branch points in bronchus images. (a2) and (b2) Branch point detection results using the proposed method.

TABLE III
PRECISION AND RECALL RATES OF DIFFERENT METHODS FOR 3D BRONCHUS IMAGE STACKS

Image Stacks	Precision (%)			Recall (%)		
	M1	M2	M3	M1	M2	M3
Bronchus1	80.00	64.52	62.50	95.24	86.96	57.14
Bronchus2	82.76	66.67	77.42	96.00	85.71	82.76
Bronchus3	88.89	78.05	84.21	94.12	91.43	80.00
Bronchus4	80.00	66.67	55.17	94.12	72.73	88.89
Bronchus5	88.89	72.73	76.19	96.97	91.43	78.05
Bronchus6	91.43	71.11	78.05	94.12	82.05	80.00
Bronchus7	82.35	66.67	43.75	93.33	66.67	70.00
Bronchus8	89.29	67.57	67.57	96.15	80.65	71.43
Bronchus9	78.95	65.22	60.00	93.75	75.00	83.33
Average	85.71	69.54	68.40	95.02	82.68	76.09

bronchus images show that the proposed method is robust to the thick structures and has a wider scope of application. The special architecture of the improved 3D U-Net and multi-scale sampling strategy of MSMV-Net is the main reason why the proposed method can reject false positives at the thick trachea area.

E. Evaluation on Retinal Blood Vessel Images

The testing retinal blood vessel images were selected from the DRIVE [44] and HRF [45] datasets. The DRIVE dataset contains 40 2D images of size 584×565 pixels while the HRF dataset contains 45 2D images of size 3504×2336 pixels, all images are coming with binary gold standard segmentation images, which were generated by a group of experts. Due to the fact that the inputs of the improved 3D-U-Net used for candidate segmentation must be three-dimensional, we converted the 2D images to 3D images by stacking and padding before testing. In Fig. 10, the branch points in segmented retinal blood vessel images are indicated by red spheres.

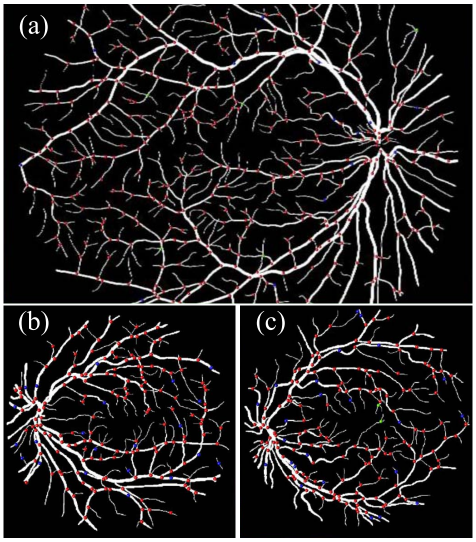

Fig. 10. Examples of 3D branch point detection results of retinal blood vessel images using the proposed method. (a) is an image from HRF database. (b) and (c) are images from the DRIVE database. The red, blue, and green spheres indicate true positive points, false positive points, and false negative points, respectively.

TABLE IV
PRECISION AND RECALL RATES OF DIFFERENT METHODS FOR RETINAL BLOOD VESSEL IMAGES

Image Stacks	Precision (%)			Recall (%)		
	M1	M2	M3	M1	M2	M3
DRIVE1	82.25	67.48	91.45	97.89	91.45	67.15
DRIVE2	93.10	69.95	96.43	95.07	90.00	60.81
DRIVE3	90.06	75.92	88.96	96.67	90.06	65.32
DRIVE4	90.63	73.60	88.41	96.67	91.19	66.51
DRIVE5	91.76	77.23	92.31	97.50	88.64	65.55
DRIVE6	92.86	81.25	86.67	97.69	91.85	66.80
DRIVE7	88.34	73.47	91.72	97.30	90.00	69.57
HRF1	92.38	88.99	94.17	97.98	95.41	56.40
HRF2	93.66	85.86	98.27	97.98	95.51	55.92
HRF3	88.10	79.74	87.26	97.37	88.10	57.28
HRF4	89.73	82.23	94.78	97.12	93.28	57.00
HRF5	90.95	79.03	96.35	95.48	92.14	55.24
HRF6	92.67	82.68	96.20	98.06	94.05	55.97
Average	90.84	79.46	93.10	97.25	92.22	59.81

However, there is a number of false positive branch points (blue spheres), especially in the areas with high curvature or close to other branches. To quantitatively evaluate the performance of our method, we compared the proposed method (M1) with the 2.5D CNNs based method (M2) and APP2 (M3). The comparison results are shown in Table IV. We can see that M1 outperforms M2 and M3, and both the precision rate

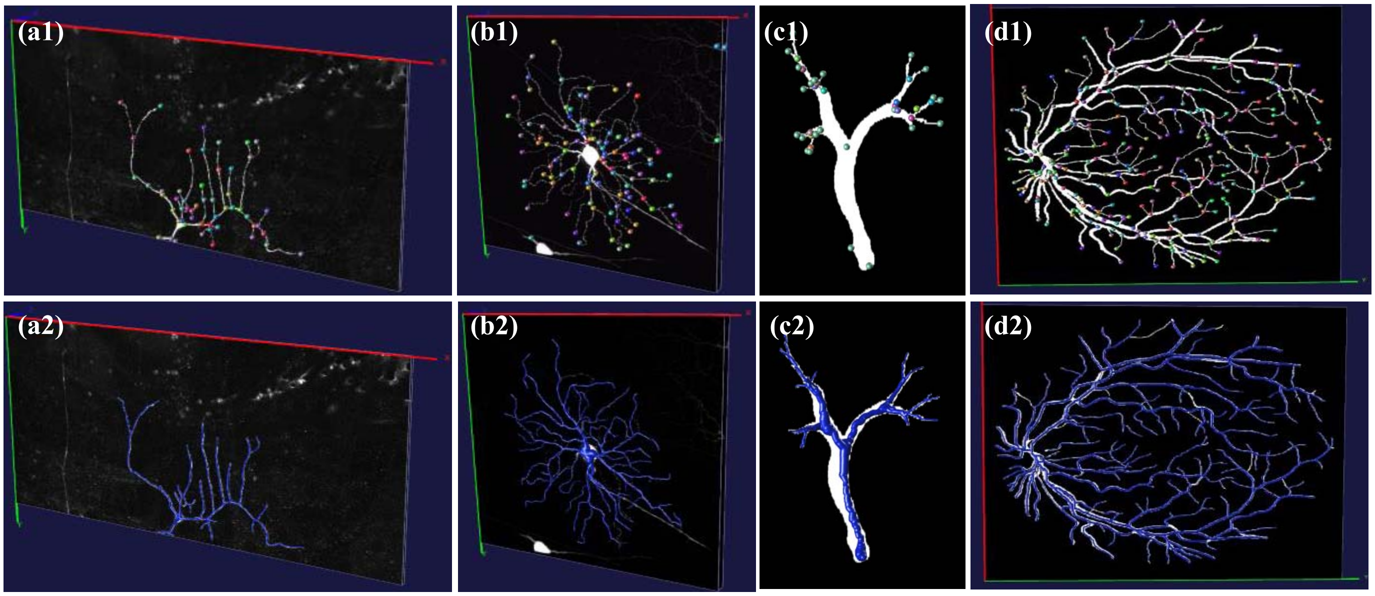

Fig. 11. The visual inspections of the tree-like structure reconstruction results. The tree-like structures are reconstructed using Vaa3D-Neuron based on the automatically detected branch points and terminations (colored spheres). (a1) and (b1) are the 3D neuron images. (c1) is a bronchus image (d1) is a retinal blood vessel. (a2)-(d2) are the corresponding reconstruction results overlaid on top of the images.

and recall rate of M1 are very high, especially the recall rate. The experimental results on retinal blood vessel images also demonstrate the effectiveness of our method for branch point detection in segmented retinal blood vessel images.

The overall average precision and recall rates of the two-level CNNs based method on all tested images^2 are shown in Table V. It demonstrates that our two-level CNNs cascade framework significantly outperforms the other two methods. In biomedical research, the detection of branch points is of significance in many applications, such as digital reconstruction of tree-like structure, which is discussed in the following section.

TABLE V
AVERAGE PRECISION AND RECALL RATES OF CNNS CASCADE METHOD

Image Stacks	Number of Images	Average Precision (%)	Average Recall (%)
3D Neuron	40	81.52	83.13
Bronchus	30	84.52	96.06
Blood Vessel	30	89.96	97.59
Average		84.95	91.35

F. Tree-Like Structure Reconstruction and Diagnosis Using the Detected Branch Points

The critical points of tree-like structures, including the terminations and branch points, which are local features that can help to determine the topology and faithfulness of the tree-like structure reconstructions [3], can be used directly in either fully-automatic or semi-automatic reconstruction methods such as the Vaa3D-Neuron [46]. The 'shortest path' algorithm is usually employed to find a smooth structure in the image voxel domain to connect one point (root) to all remaining critical points with the least 'cost' [6].

Illustrative examples of the tree-like structure reconstruction results based on terminations and branch points together are shown in Fig. 11. We also compared the reconstruction results based on only terminations with the results based on branch points and terminations together, as depicted in Fig. 12. From Fig. 12. (a3), and Fig. 12. (b3), we can see that the addition of branch points can help to trace the missing branches which are resulted from the loops between the terminations. Therefore, the detection of branch points is extremely helpful to tree-like structure reconstruction [24].

And another critical problem for the existing reconstruction methods is to trace dense neurites with inferences from nearby neurite. These neurites may interconnect with each other and form crossover regions. If the locations of crossover points can be indicated, that would be very much helpful to the reconstruction of dense neurons. Although we focused on branch points detection in this work, the cascade model has the potential to detect crossover points by adding multi-neuron images containing more crossover points, from new datasets.

The detected branch points can also help to diagnose the reconstruction results. For most existing reconstruction methods, the procedure of reconstructing the morphology of the tree-like structures in biomedical images is actually the detection of a series of nodes with parent-child relationship, as shown in Fig. 13. (a)-(b). In the reconstruction results of APP2, over- reconstruction usually occurs near a node that has two parents or children but no real branch point nearby, as shown in Fig. 13. (c). On the contrary, the phenomenon where the reconstructed morphology does not bifurcate near a real branch point usually represents missing branches, as shown in Fig. 13. (d). Therefore, automatic detection of branch points can effectively help to diagnose the

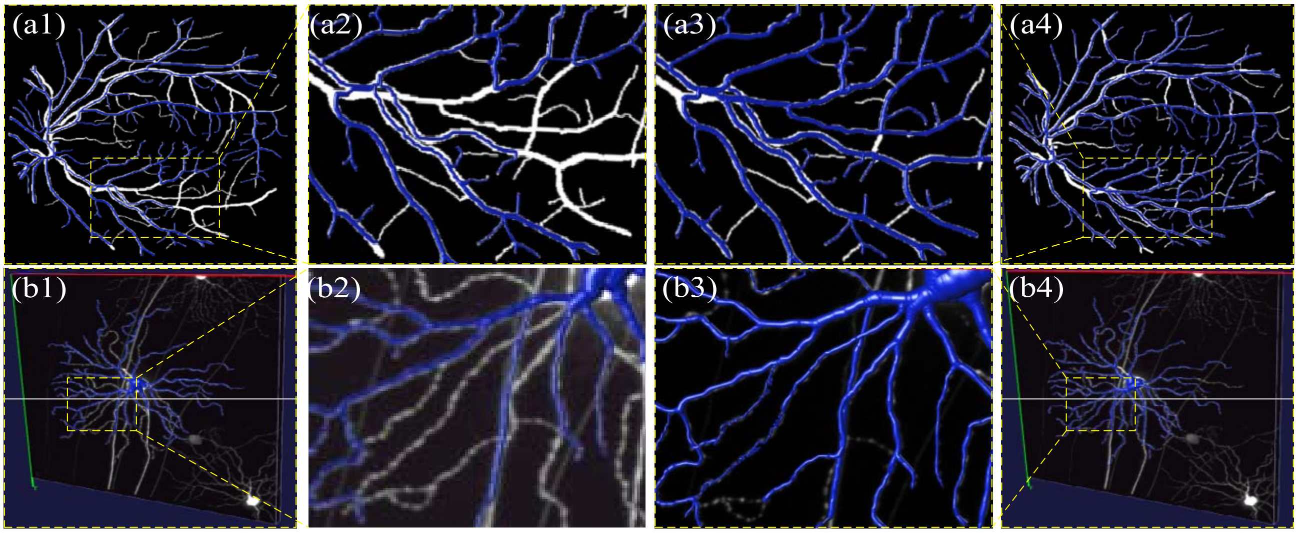

Fig. 12. Visual inspection of the tree-like structure reconstruction results using Vaa3D-Neuron. (a1) and (b1) are reconstruction results based on only terminations. (a4) and (b4) are reconstruction results based on branch point and terminations together. The details of the zoomed-in view in two representative image areas (yellow boxes) are shown in (a2), (a3), (b2), and (b3).

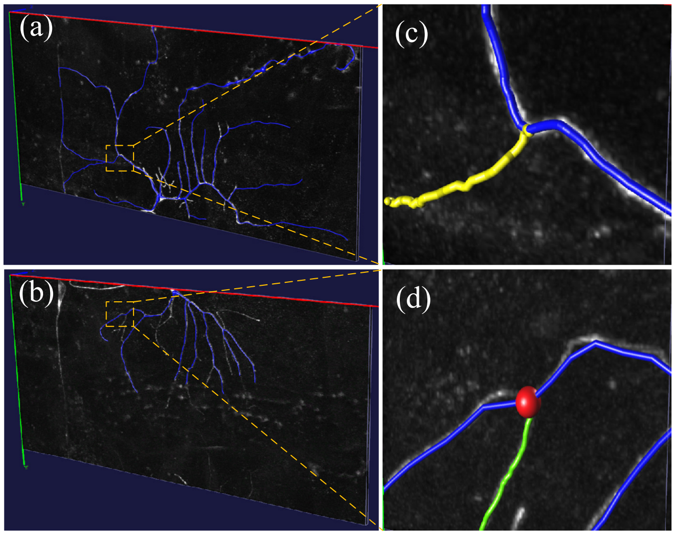

Fig. 13. Diagnosing the reconstruction results using branch points. (a) and (b) are reconstruction results of APP2. (c) is an example of over-reconstruction (yellow segments). (d) is an example of missing branch (green segments), the red sphere is a detected branch point.

reconstruction results, whether it is automatically traced or manually traced.

IV. Conclusions

In this paper, we propose a two-level CNNs cascade framework for branch point detection in biomedical images. An improved 3D U-Net model is employed to segment candidate regions, which is more effective and easier to train and test than local fixed patch-based approaches. By adding multi-scale sampling strategy, using the multi-view CNNs architecture and an appropriate feature fusion method in MSMV-Net, we can separate the true branch points form false positives effectively. The experimental results on multiple datasets show that our two-level CNNs cascade framework achieves higher detection accuracy than other branch point detection methods and outperforms the branch point detection results inferred from existing neuron reconstruction methods. The reconstruction results demonstrate that the joint use of branch points and terminations can improve the performance of the existing reconstruction methods compared to using terminations only.

References

[1] E. Meijering, "Neuron tracing in perspective," *Cytometry A*, vol. 77, no. 7, pp. 693–704, 2010.

[2] 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.

[3] 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.

[4] 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.

[5] H. Peng, F. Long, and G. Myers, "Automatic 3D neuron tracing using all-path pruning," *Bioinformatics*, vol. 27, no. 13, pp. i239–i247, Jul. 2011.

[6] 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, Apr. 2017.

[7] M. Radojevic and E. Meijering, "Automated neuron tracing using probability hypothesis density filtering," *Bioinformatics*, vol. 33, no. 7, pp. 1073–1080, 2017.

[8] 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.

[9] G. González, E. Türetken, F. O. Fleuret, and P. Fua, "Delineating trees in noisy 2D images and 3D image-stacks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2799–2806.

[10] H. Peng, Z. Ruan, D. Atasoy, and S. Sternson, "Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model," *Bioinformatics*, vol. 26, no. 12, pp. I38–I46, Jun. 2010.

[11] Y. Wang, A. Narayanaswamy, C. Tsai, and B. Roysam, "A broadly applicable 3-D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, nos. 2–3, pp. 193–217, 2011.

[12] Y. Al-Kofahi, N. Dowell-Mesfin, C. Pace, W. Shain, J. N. Turner, and B. Roysam, "Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images," *Cytometry A*, vol. 73, no. 1, pp. 36–43, 2008.

[13] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2008, pp. 650–663.

[14] G. Azzopardi and N. Azzopardi, "Trainable COSFIRE filters for keypoint detection and pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 490–503, Feb. 2013.

[15] L. Rada, E. Erdil, A. OzgurArgunsah, D. Unay, and M. Cetin, "Automatic dendritic spine detection using multiscale dot enhancement filters and SIFT features," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 26–30.

[16] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. ACM Int. Conf. Multimedia (ACM)*, Sep. 2007, pp. 357–360.

[17] I. Sipiran and B. Bustos, "Harris 3D: A robust extension of the harris operator for interest point detection on 3D meshes," *Vis. Comput.*, vol. 27, no. 11, pp. 963–976, 2011.

[18] M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1704–1711.

[19] M. Liu, H. Peng, A. K. Roy Chowdhury, and E. W. Myers, "3D neuron tip detection in volumetric microscopy images," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2011, pp. 366–371.

[20] M. Liu, R. Gong, W. Chen, and H. Peng, "3D neuron tip detection in volumetric microscopy images using an adaptive ray-shooting model," *Pattern Recognit.*, vol. 75, pp. 263–271, Mar. 2018.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[23] H.-C. Shin *et al.*, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[24] M. Radojevié, I. Smal, and E. Meijering, "Automated neuron morphology reconstruction using fuzzy-logic detection and Bayesian tracing algorithms," in *Proc. IEEE 12th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2015, pp. 885–888.

[25] 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)*, Dec. 2018, pp. 443–448.

[26] W. Zhang *et al.*, "Deep convolutional neural networks for multi-modality isointense infant brain image segmentation," *NeuroImage*, vol. 108, pp. 214–224, Mar. 2015.

[27] Q. Dou *et al.*, "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1182–1195, May 2016.

[28] M. J. J. P. van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sánchez, "Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1273–1284, May 2016.

[29] Z. Yan, X. Yang, and K.-T. Cheng, "Joint segment-level and pixel-wise losses for deep learning based retinal vessel segmentation," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 9, pp. 1912–1923, Sep. 2018.

[30] P. Hattikatti, "Texture based interstitial lung disease detection using convolutional neural network," in *Proc. Int. Conf. Big Data, IoT Data Sci. (BID)*, Dec. 2017, pp. 18–22.

[31] Y. Tan, H. Luo, X. Wang, and M. Liu, "Convolutional neural network cascade based neuron termination detection in 3D image stacks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4048–4052.

[32] A. A. A. Setio *et al.*, "Pulmonary nodule detection in CT images: False positive reduction using multi-view convolutional networks," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1160–1169, May 2016.

[33] X. Liu, F. Hou, H. Qin, and A. Hao, "Multi-view multi-scale CNNs for lung nodule type classification from CT images," *Pattern Recognit.*, vol. 77, pp. 262–275, May 2018.

[34] C. Dai *et al.*, "Automated detection of lung nodules in CT images with 3D convolutional neural networks," in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Aug. 2018, pp. 55–59.

[35] M. Ghafoorian *et al.*, "Deep multi-scale location-aware 3D convolutional neural networks for automated detection of lacunes of presumed vascular origin," *NeuroImage, Clin.*, vol. 14, pp. 391–399, Feb. 2017.

[36] G. Tetteh *et al.*, "DeepVesselNet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes," 2018, *arXiv:1803.09340*. [Online]. Available: https://arxiv.org/abs/1803.09340

[37] K. Kamnitsas *et al.*, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, Feb. 2017.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent (MICCAI)*, 2015, pp. 234–241.

[39] G. Wang *et al.*, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Trans. Med. Imag.*, vol. 37, no. 7, pp. 1562–1573, Jul. 2018.

[40] A. Agarwal *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467

[41] 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, Apr. 2013.

[42] H. Peng, E. Meijering, and G. A. Ascoli, "From diadem to BigNeuron," *Neuroinformatics*, vol. 13, no. 3, pp. 259–260, 2015.

[43] B. van Ginneken *et al.*, "Comparing and combining algorithms for computer-aided detection of pulmonary nodules in computed tomography scans: The ANODE09 study," *Med. Image Anal.*, vol. 14, no. 6, pp. 707–722, 2010.

[44] J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. V. Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[45] J. Odstrcilik *et al.*, "Retinal vessel segmentation by improved matched filtering: Evaluation on a new high-resolution fundus image database," *IET Image Process.*, vol. 7, no. 4, pp. 373–383, Jun. 2013.

[46] 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.

[47] S. Hang, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Dec. 2015, pp. 945–953.