

# DeepBranchTracer: A Generally-Applicable Approach to Curvilinear Structure Reconstruction Using Multi-Feature Learning

Chao Liu<sup>1,2</sup>, Ting Zhao<sup>3</sup>, Nenggan Zheng<sup>1, 2, 4, 5\*</sup>

<sup>1</sup>Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou, Zhejiang, China

<sup>2</sup>College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China

<sup>3</sup>Independent Researcher

<sup>4</sup>State Key Lab of Brain-Machine Intelligence, Zhejiang University, Hangzhou, Zhejiang, China

<sup>5</sup>CCAI by MOE and Zhejiang Provincial Government(ZJU), Hangzhou, Zhejiang, China  
supermeliu@zju.edu.cn, tingzhao@gmail.com, zng@cs.zju.edu.cn

## Abstract

Curvilinear structures, which include line-like continuous objects, are fundamental geometrical elements in image-based applications. Reconstructing these structures from images constitutes a pivotal research area in computer vision. However, the complex topology and ambiguous image evidence render this process a challenging task. In this paper, we introduce DeepBranchTracer, a novel method that learns both external image features and internal geometric characteristics to reconstruct curvilinear structures. Firstly, we formulate the curvilinear structures extraction as a geometric attribute estimation problem. Then, a curvilinear structure feature learning network is designed to extract essential branch attributes, including the image features of centerline and boundary, and the geometric features of direction and radius. Finally, utilizing a multi-feature fusion tracing strategy, our model iteratively traces the entire branch by integrating the extracted image and geometric features. We extensively evaluated our model on both 2D and 3D datasets, demonstrating its superior performance over existing segmentation and reconstruction methods in terms of accuracy and continuity.

## Introduction

Curvilinear structure is a line-like, elongated object with varying intensities compared to their surroundings. The extraction of curvilinear structures aims to quantitatively identify these objects from image data, facilitating statistical analyses of branch connections and distribution. This represents a pivotal research area in computer vision, particularly for neuron branches and blood vessels in medical images (Meijering 2010; Soomro et al. 2019), and roads in aerial images (Chen et al. 2022). However, despite decades of research efforts dedicated to automating this process, extracting curvilinear structures in occlusion and ambiguous topology images still remains a challenging task.

Over the past few years, curvilinear structure extraction has always been addressed as a segmentation-based problem. As an essential segmentation tool, convolutional neural networks (CNNs) can automatically extract image features from diverse images (Li et al. 2017; Long, Shelhamer, and Darrell 2015). However, unlike typical segmentation tasks,

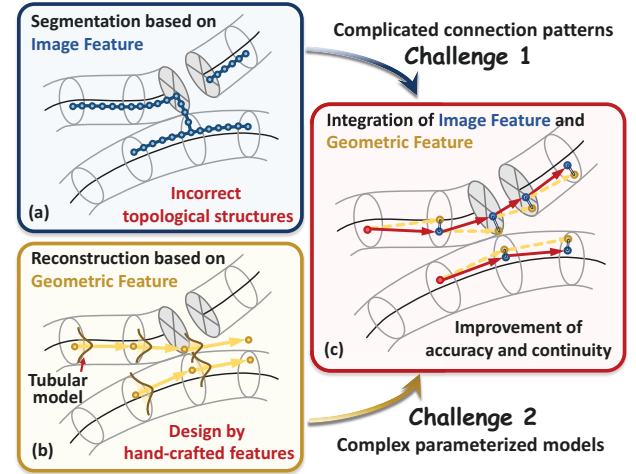


Figure 1: Motivation. Segmentation methods focus on learning image features but tend to misinterpret two adjacent branches as an entity. Reconstruction methods capture the geometric features of branches but rely on complex parameterized models. The integration of these two features offers the potential to improve the continuity of results.

curvilinear structure extraction places heightened emphasis on the continuity and connectivity of the target, rather than just its contour. Several studies have incorporated the prior structural constraints into the network architectures or loss functions to enhance their effectiveness (Qi et al. 2023; Zhou et al. 2021; Liu et al. 2022a). Nevertheless, a foundational challenge remains across these segmentation-based methods. When confronted with two adjacent or tangled branches, segmentation-based methods often tend to misinterpret them as a single entity, and will constitute a crucial topological error, as depicted in Fig. 1(a).

In response to these challenges, reconstruction-based methods have emerged to explicitly model the topological representation of curvilinear structures, as shown in Fig. 1(b). These methods encompass both external image features and internal geometric features, and can be categorized into two groups based on their reconstruction processes: **global-first** approaches and **local-first** approaches.

\*Corresponding author.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Global-first approaches, as a variant of segmentation-based methods, extract nodes or segments from the whole image perspective to produce a skeleton. These top-down approaches, such as graph optimization (Türetken et al. 2016), skeletonization (Yuan et al. 2009) and signal enhancement (Yang et al. 2021), can efficiently explore weak branches by merging the potential edges into networks. However, a major drawback of these approaches is that they have an increased likelihood of generating *spurious branches* due to mistaken seed points from noise. Therefore, a post-processing pruning algorithm or labor-intensive human correction is necessary to generate the definitive reconstruction (Peng et al. 2011).

Local-first approaches start from a reasonable seed point to explore the potential branches. As a conservative strategy, these approaches prioritize local information and cease execution when image signals become indistinct, thereby avoiding the error of spurious branches. Thus, local-first approaches become a promising way for addressing complex topology challenges in large-scale image applications, such as neuron tracing or city road extraction. Nonetheless, these approaches often rely on geometric models designed with hand-crafted features, such as tubular template (Zhao et al. 2011) and ray-shooting model (Ming et al. 2013), and the incorporation of these models as modules into a deep network for training is challenging. Local-first approaches also place significant reliance on the accuracy of the global pre-segmentation mask to decide whether to continue tracing, otherwise the weak branches could be omitted in the results.

To overcome the above limitations of the local-first approaches, we proposed a novel reconstruction model called DeepBranchTracer, involving a curvilinear structure feature learning network (CSFL) and a multi-feature fusion tracing strategy (MFT). Firstly, to address the challenge of integrating geometric models into a deep network, we formulate a mathematical framework for describing curvilinear structures and investigate the geometric parameters that necessitate optimization. Secondly, to expand the interest region of local-first approaches, we propose a U-shaped network with four subtasks blocks to learn both the external image feature and internal geometric continuous feature in an integrated CSFL network. For the image features, the centerline serves to fine-tune the branch positions, while the boundary determines whether to stop a tracing process. For the geometric features, a sequence learning module is designed for direction and radius estimation blocks to capture smoothness and continuity of the branch signals. Finally, we developed an MFT strategy to explore the curvilinear structure iteratively based on the branch multiple features extracted above.

Our main contributions can be summarized as follows:

- We formulate the problem of curvilinear structure extraction as an explicit geometric parameter estimation task for deep networks, eliminating the need for complex geometric parametric models based on hand-craft features.
- DeepBranchTracer involves the CSFL network to learn both internal geometric feature and external image feature of branches, and combines the MFT strategy, enabling an iterative tracing process for finalizing the re-

construction.

- To our best knowledge, DeepBranchTracer is the first work for curvilinear structure reconstruction for both 2D and 3D images. With extensive experiments on five various datasets, our model has achieved better accuracy and continuity compared to other state-of-the-art (SOTA) segmentation and reconstruction methods.

## Related Work

### Segmentation-based Methods

Various segmentation-based methods for curvilinear structure have been proposed to achieve better results by adjusting model backbones or feature learning modules. As a prominent segmentation architecture, U-Net (Ronneberger, Fischer, and Brox 2015) has inspired a diverse of variant network models for curvilinear structure segmentation, including the integration of ResBlock (Milletari, Navab, and Ahmadi 2016), feature aggregation (Liu et al. 2022b), attention mechanisms (Schlemper et al. 2019), among others. To improve the continuity of the results, the deformable convolution (Dai et al. 2017) and its derivatives (Yang et al. 2022; Qi et al. 2023) were proposed to permit the network to dynamically adjust the receptive field, facilitating the learning of geometric object features. Additionally, SGCN (Zhou et al. 2021) employed a split depth-wise separable graph convolutional network to capture global contextual information in channel and spatial features. Nonetheless, a foundational challenge remains across these segmentation-based methods. When input image evidence contains fragmented or obstructed branches, networks struggle to learn structural features effectively, and require a post-processing to repair the broken segmentation results.

### Reconstruction-based Methods

The reconstruction-based methods can be divided into two categories. As an example of the global-first approaches, (Türetken et al. 2016) introduced an integer programming technique to identifying the optimal subset of paths, subject to topological constraints that eliminate implausible solutions. On the basis of this, (Mosinska, Koziński, and Fua 2019) trained a deep network to segment and evaluate the potential paths as a data-driven method. In contrast, local-first approaches start from a set of seed points and employ a greedy tracing methodology guided by geometric template fitting score. These methods are effective in certain applications but rely on manually crafted features, such as voxel scooping (Rodriguez et al. 2009), ray shooting (Ming et al. 2013), and tubular fitting (Zhao et al. 2011), making them difficult to choose tuning parameters when faced with different datasets. The end-to-end reconstruction methods SPE (Chen et al. 2021) and RoadTracer (Bastani et al. 2018) have also been proposed to iteratively estimate tracing directions. SPE employs a spherical core to transform the 3D stacks of neuron images into 2D patches and also leads to the loss of texture information in the original images, limiting its applicability to other fields beyond microscope grayscale images. RoadTracer constructs a CNN-based model that iteratively builds a road network graph using direction estimation

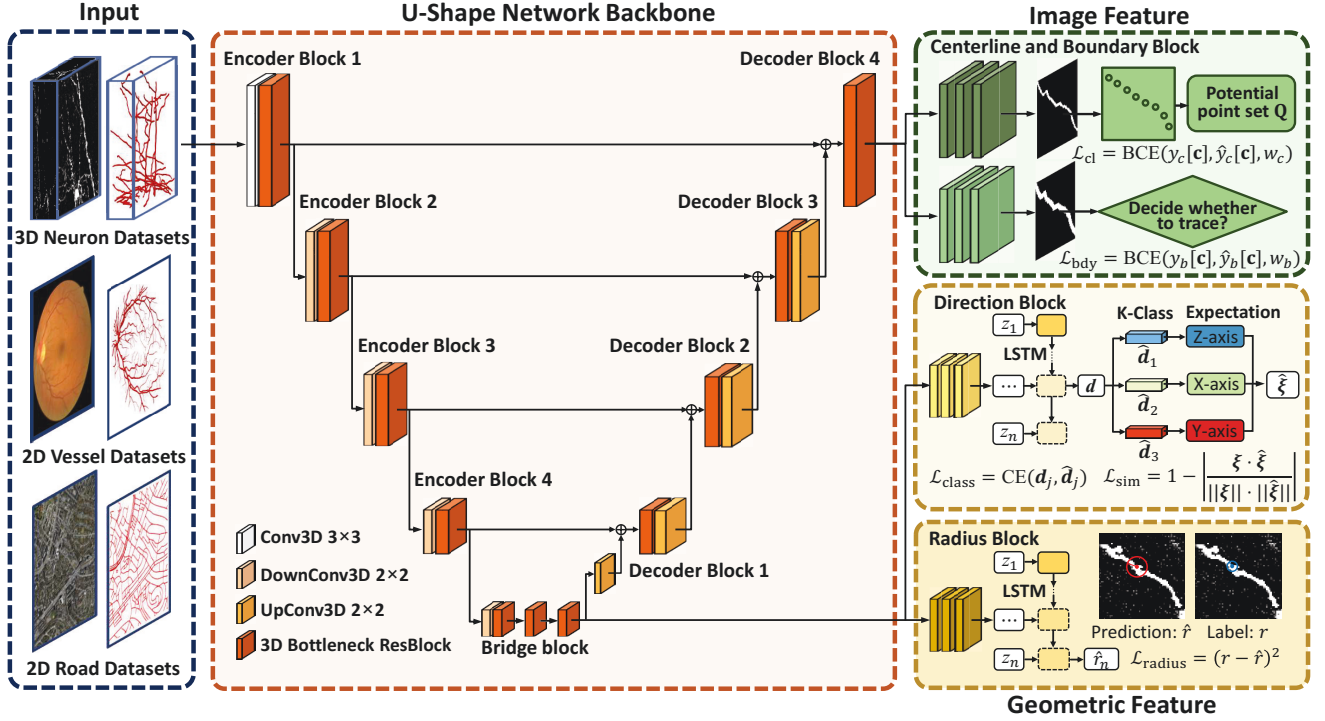


Figure 2: Architecture of the CSFL network. The network integrates four sub-task blocks in a U-shaped network to extract both geometric features (including radius and direction blocks) and image features (including centerline and boundary blocks).

and a decision function. However, it lacks a direction repair module, which will always cause an early stop in the tracing process in case of an unexpected prediction error.

Compared with earlier tracing methods, our model eliminates the need for complex parameterized geometric models, seamlessly integrates image features and geometric features to enhance the tracing capabilities, particularly for elongated branches.

## Method

In this section, we formulate the problem of curvilinear structure reconstruction as the task of geometric parameters estimation, including radius, direction and position. Subsequently, we introduce our DeepBranchTracer model, which leverages a CSFL network and four sub-task blocks to acquire insights from both image and geometric features. Finally, we detail the incorporation of learned branch attributes into the tracing process through the MFT strategy.

### Problem Formulation

A curvilinear structure can be represented as a continuous solid whose axis is a 3-D space curve. This curve is a set of points  $C \subset \mathbb{R}^3$  with the property that for every  $c \in C$  there exists a smooth function  $\{c(t) = (x(t), y(t), z(t))^T, t \in (a, b)\}$ , where  $c(t)$  is the centerline of the curvilinear structure. Each point also has a property of radius  $r(t)$  to form a circular cross section. Along the centerline, we can integrate these circular cross sections and form a solid curvilinear structure  $S(t)$  to fit the branch object in image  $I$ . To fa-

cilitate the calculation, we approximated the continuous centerline as  $m$  pieces of segments, and the curvilinear structure  $S(t)$  can be represented by the discrete cylinders  $S_i(t)$ :

$$S(t) = \sum_{i=1}^m S_i(t), \quad (1)$$

and each cylinder can be represented as:

$$S_i(t) \approx \begin{cases} (1-\gamma(t))c(t_i) + \gamma(t)c(t_{i+1}) \\ \quad + R(t_i)D(t_i)O(u) \\ 0 \end{cases}, \text{ if } t \in (t_i, t_{i+1}) \\ \text{, otherwise} \quad (2)$$

where

$$\gamma(t) = \frac{t - t_i}{t_{i+1} - t_i}, \quad (3)$$

where  $R(t_i) = \varphi \cdot r(t_i)$  represents the space within the cylinder radius, and  $\varphi \in (0, 1]$ .  $D(t_i) = (\nu(t_i), \zeta(t_i))$  are two orthogonal vectors to form the normal plane of the centerline with reference to the unit direction vector  $\xi(t_i)$ , and can be obtained based on the Frenet frame, which is elaborated in the Appendix.  $O(u) = (\cos u, \sin u)^T$  is the parametric equation of a circle, and  $u \in (0, 2\pi]$ .

Thus, the problem of accurately reconstructing a curvilinear structure can be viewed as an iterative process involving the estimation of the position point  $c(t_i)$ , its radius  $r(t_i)$ , and direction  $\xi(t_i)$  to determine the subsequent point  $c(t_{i+1})$ :

$$c(t_{i+1}) = c(t_i) + r(t_i)\xi(t_i). \quad (4)$$

As a constraint condition, all the point  $\mathbf{c}(t_i)$  should adhere to the condition  $\mathbf{c}(t_i) \in S(t)$  to prevent deviation from the centerline. Finally, this process yields the ultimate solid  $S(t)$  as the reconstruction result. Here, we propose the overall loss function and look for the suitable  $r^*$ ,  $\xi^*$  and  $\mathbf{c}^*$ :

$$\min \sum_{i=1}^m (\mathcal{L}_{\text{geo}}(r(t_i), \xi(t_i), I[\mathbf{c}(t_{i-l:i})]) + \mathcal{L}_{\text{img}}(y_b[\mathbf{c}(t_i)], y_c[\mathbf{c}(t_i)], I[\mathbf{c}(t_i)])) \quad (5)$$

where  $I[\mathbf{c}(t_i)]$  is the image at the point  $\mathbf{c}(t_i)$ ,  $I[\mathbf{c}(t_{i-l:i})]$  represents the the image patches from  $l$  historical points, and  $y_b, y_c$  are the segmentation labels of image.  $\mathcal{L}_{\text{geo}}$  represents the geometry loss associated with radius and direction estimation, while  $\mathcal{L}_{\text{img}}$  signifies the image loss employed to incorporate image features during the tracing process. We will describe these two loss terms in more detail below.

### Curvilinear Structure Feature Learning Network

To estimate the above branch properties, we employ the CSFL network to extract the image and geometric features using four sub-task blocks: centerline and boundary blocks, direction and radius blocks, as depicted in Figure 2.

**U-shaped Backbone.** The U-shaped backbone comprises three main components: an encoder, a bridge, and a decoder. The encoder is implemented as a 4-level bottleneck ResBlock, while the bridge exhibits a broader 3-level block compared to the original U-Net model. This enhanced configuration allows for the extraction of more comprehensive high-level semantic information, and can be represented as  $f_{\phi_E} : I[\mathbf{c}(t_i)] \rightarrow z_e(t_i)$ . The output of the bridge serves as the input for both the direction and radius block, enhancing the provision of geometric features. The decoder exhibits an analogous architecture to the encoder, with the goal of upsampling the encoded features to match the original resolution of the input image, and can be represented as  $f_{\phi_D} : z_e(t_i) \rightarrow z_d(t_i)$ . It generates inputs for the centerline and boundary blocks.

**Geometric Feature.** In the previous work, several methods (Chen et al. 2021; Bastani et al. 2018) have employed CNNs to predict geometric features within a local context. However, it is widely observed that points along a continuous branch often manifest similar direction and radius values in comparison to their neighboring points. To capture this natural attribute, we incorporate the sequential learning module into the radius and direction blocks, facilitating the capture of branch smoothness and continuity. The feature maps derived from historical continuous points are fed as input to predict the geometric features of the successor point.

We have devised the direction block using an LSTM module, serving as the mapping function  $f_{\theta_d} : z_e(t_{i-l:i}) \rightarrow d(t_i)$ . The direction block is composed of two losses, which is the classification loss of bins denoted as  $\mathcal{L}_{\text{class}}$  and the similarity loss of direction referred to as  $\mathcal{L}_{\text{sim}}$ . The task of  $K$ -class classification for direction estimation is designed to ascer-

tain the bin to which the vector component belongs:

$$\mathcal{L}_{\text{class}} = \frac{1}{J} \sum_{j=1}^J \text{CE}(\mathbf{d}_j(t_i), \hat{\mathbf{d}}_j(t_i)), \quad (6)$$

where  $J$  is the number of vector component. In 3D space,  $J$  equals 3, and  $\mathbf{d}(t_i)$  comprises components along the  $X$ ,  $Y$ , and  $Z$  axes, denoted as  $[\mathbf{d}_x(t_i), \mathbf{d}_y(t_i), \mathbf{d}_z(t_i)]$ .

However, the adjacent bins of the classification are interconnected in the direction estimation task. For instance, when the direction label is  $[0, 1, 0, 0, 0]$ , the predicted probability of  $[0, 0, 0.9, 0, 0]$  is expected to be superior to the probability of  $[0, 0, 0, 0.9, 0]$ , even though they have the same classification loss. Therefore, from a geometric perspective, in order to obtain the fine-grained direction predictions, we introduce a cosine similarity loss that computes the disparity between the two direction vectors:

$$\mathcal{L}_{\text{sim}} = 1 - \left| \frac{\xi(t_i) \cdot \hat{\xi}(t_i)}{\|\xi(t_i)\| \cdot \|\hat{\xi}(t_i)\|} \right|, \quad (7)$$

and for each vector component of  $\hat{\xi}(t_i)$ :

$$\hat{\xi}_j(t_i) = \sum_{k=1}^K \text{softmax}(\hat{\mathbf{d}}_j(t_i))_k \cdot A_k \quad (8)$$

where  $K$  is the number of classes, and  $A_k$  is the expectation of each bin. The direction  $\hat{\xi}(t_i)$  is computed by taking the expectation of the binned output for each vector component. Specifically, a direction can be represented as the positive  $\xi$  and negative  $-\xi$  vectors of the same value but diff sign. When the cosine similarity between the predicted and ground truth vectors equals 1 or -1, the  $\mathcal{L}_{\text{sim}}$  will be 0, signifying a higher direction similarity. The final loss of direction block represented as:

$$\mathcal{L}_{\text{direction}} = \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{sim}}. \quad (9)$$

Similarly, the radius block utilizes the same sequential learning module and is designed as the mapping of  $f_{\theta_r} : z_e(t_{i-l:i}) \rightarrow \hat{r}(t_i)$ . To train this module, the mean-squared error loss is employed:

$$\mathcal{L}_{\text{radius}} = (r(t_i) - \hat{r}(t_i))^2. \quad (10)$$

Combined the direction block and the radius block, the final geometry loss is integrated as:

$$\mathcal{L}_{\text{geo}} = \lambda_d \mathcal{L}_{\text{direction}} + \lambda_r \mathcal{L}_{\text{radius}}, \quad (11)$$

where  $\lambda_d$  and  $\lambda_r$  are the weights to balance these two losses.

**Image Feature.** The purpose of centerline extraction is to provide the potential successor points based on the image features. Here, we consider the problem of centerline extraction as a binary segmentation task. However, the centerline of curvilinear structure is often thin and line-like, leading to an imbalance of positive and negative data in the training dataset. To address this issue, we employ a weighted binary cross entropy loss function, which assigns a higher cost to

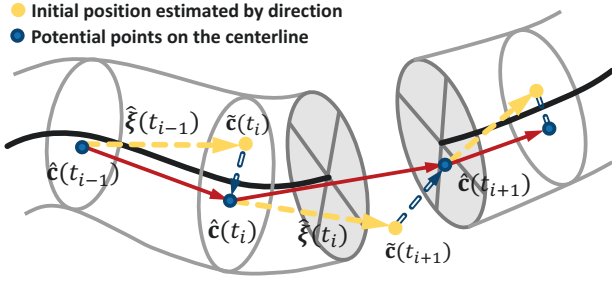


Figure 3: MFT strategy. Step 1: Find an initial point  $\tilde{c}(t_i)$  based on the geometric features of direction and radius. Step 2: From the point set  $\mathbf{Q}$  of the surrounding centerline, locate the most potential point  $\hat{c}(t_i)$  to adjust the position. Step 3: Determine whether to trace the successor point based on the boundary probability  $\hat{y}_b(\hat{c}(t_i))$ .

the foreground branch in order to improve the performance. The centerline loss  $\mathcal{L}_{cl}$  is formulated as:

$$\mathcal{L}_{cl} = \text{BCE}(y_c[\mathbf{c}(t_i)], \hat{y}_c[\mathbf{c}(t_i)], w_c), \quad (12)$$

where  $y_c[\mathbf{c}(t_i)]$  and  $\hat{y}_c[\mathbf{c}(t_i)]$  indicate the ground truth and probability map of centerline,  $w_c$  is the weight for foreground of centerline. Based on the Eq. (2), we set the radius  $r$  of all points to 1 to generate the centerline label.

The boundary block determines whether to continue tracing in the next step. Previous researches (Chen et al. 2021; Bastani et al. 2018) employed a fully connected layer classifier to ascertain the probability of the present location belonging to either the foreground *branch* or background *noise*. Nonetheless, this approach consolidates all feature maps into a singular vector, resulting in the loss of position information within the image. Here, we approach the tracing decision as a boundary segmentation task:

$$\mathcal{L}_{bdy} = \text{BCE}(y_b[\mathbf{c}(t_i)], \hat{y}_b[\mathbf{c}(t_i)], w_b), \quad (13)$$

where  $\hat{y}_b[\mathbf{c}(t_i)]$  is the probability map of the boundary,  $w_b$  is the weight for foreground of boundary. We set the radius  $r$  of all points to  $r_b$  to generate the boundary label. When  $r_b$  is a large value, the tracing algorithm tends to adopt an aggressive strategy; conversely, it becomes conservative.

The overall loss function of image feature can be defined as follows:

$$\mathcal{L}_{img} = \lambda_c \mathcal{L}_{cl} + \lambda_b \mathcal{L}_{bdy}, \quad (14)$$

where  $\lambda_c$  and  $\lambda_b$  are the weights to balance the two losses.

### Multi-Feature Fusion Tracing Strategy

Our model utilizes an MFT strategy, combining both internal geometric features and external image features to trace the branch iteratively, as shown in Fig. 3.

Starting from a seed point  $\mathbf{c}(t_i)$ , the module traces the branch in two steps. For the first step, we estimate the radius  $\hat{r}(t_i)$  and direction  $\hat{\xi}(t_i)$  of the current point to find an initial successor point  $\tilde{c}(t_{i+1})$  based on the Eq. (4). However, due to low quality and ambiguity of image, the position of the  $\tilde{c}(t_{i+1})$  is always low confidence. If our model follows this

### Algorithm 1: Multi-Feature Fusion Tracing Strategy

**Input:** Seed point position:  $\mathbf{c}(t_1)$ , Input image  $I$

**Output:** Reconstructed branch:  $B$

```

1: Initialize  $B$  as the empty point set  $B \leftarrow \emptyset$ , tracing action
    $a = \text{continue}$ ,  $i \leftarrow 1$ 
2: while  $a = \text{continue}$  do
3:   // Predict branch and direction parameters.
4:    $\hat{y}_c[\mathbf{c}(t_i)], \hat{y}_b[\mathbf{c}(t_i)], \hat{r}(t_i), \hat{\xi}(t_i) \leftarrow \text{Network}(I[\mathbf{c}(t_i)])$ 
5:   // Estimate initial point by direction.
6:    $\tilde{c}(t_{i+1}) = \mathbf{c}(t_i) + \hat{r}(t_i)\hat{\xi}(t_i)$ 
7:   // Adjust position based on centerline.
8:    $\hat{c}(t_{i+1}) = \arg \min_{\mathbf{c}(q) \in \{\mathbf{Q} | \hat{y}_c[\mathbf{c}(t_i)] > T_c\}} \|\tilde{c}(t_{i+1}) - \mathbf{c}(q)\|^2$ 
9:   if  $\hat{y}_b(\hat{c}(t_{i+1})) > T_b$  and  $\hat{c}(t_{i+1})$  is not in  $B$  then
10:     $\mathbf{c}(t_{i+1}) \leftarrow \hat{c}(t_{i+1})$ 
11:     $B \leftarrow B \cup \{\mathbf{c}(t_{i+1})\}$ 
12:     $i \leftarrow i + 1$ 
13:   else
14:      $a = \text{stop}$ 
15:   end if
16: end while
17: return  $B$ 

```

decision, it may deviate from the centerline. Thus, in the second step, our goal is to enhance the accuracy of the branches by incorporating the centerline feature. This involves identifying the closest point  $\hat{c}(t_{i+1})$  from the potential successor point set  $\mathbf{Q}$  to current point  $\tilde{c}(t_{i+1})$  as the final decision:

$$\hat{c}(t_{i+1}) = \arg \min_{\mathbf{c}(q) \in \{\mathbf{Q} | \hat{y}_c[\hat{c}(t_i)] > T_c\}} \|\tilde{c}(t_{i+1}) - \mathbf{c}(q)\|^2, \quad (15)$$

where  $T_c$  is the threshold of the probability map  $\hat{y}_c[\hat{c}(t_i)]$ . Points within the centerline probability map exhibiting values exceeding  $T_c$  are included within the potential successor point set  $\mathbf{Q}$ .

Furthermore, we utilize two stopping criteria for the tracing process. The first criterion involves checking if the boundary probability of the successor point  $\hat{c}(t_{i+1})$  falls below a threshold  $T_b$ . The second criterion is triggered when the successor point intersects with a previously traced region. The detail of the tracing strategy is laid out in Algorithm 1.

## Experiments

We carried out ablation experiments of our model DeepBranchTracer in order to evaluate the effects of different model settings, including the image feature, the geometric feature, and the LSTM sequence learning module. We also quantified the performance of our approach on both 2D and 3D datasets. The results show that our proposed model achieved better results compared with the SOTA segmentation-based and reconstruction-based models.

### Dataset

We conducted experiments on five popular datasets contain 2D and 3D images, consisting of Massachusetts Roads



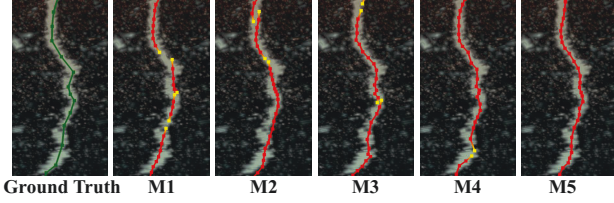


Figure 4: From left to right, with the integration of centerline and geometric features, the reconstruction branch of M5 achieves improved smoothness and continuity. The yellow dots represent the termination points of the broken branches.

	$\mathcal{L}_{cl}$	$\mathcal{L}_{geo}$	LSTM	PE	RE	SSD-F1	Length-F1	ABL
M1		✓		1.879	0.281	0.502	0.704	48.07
M2		✓	✓	1.816	0.270	0.519	0.701	57.04
M3	✓			0.835	0.264	0.845	0.893	128.65
M4	✓	✓		1.021	0.265	0.893	0.890	167.51
M5	✓	✓	✓	<b>0.775</b>	<b>0.262</b>	<b>0.910</b>	<b>0.911</b>	<b>198.33</b>

Table 1: Ablation study for geometric and image feature learning modules based on fMOST-VTA dataset.

Dataset (Mnih 2013), DRIVE (Staal et al. 2004) and CHASE\_DB1 (Carballal et al. 2018) retina datasets, 3D DIADEM-CCF (Brown et al. 2011) and fMOST-VTA (Li et al. 2010; Gong et al. 2013) neuron datasets. Specifically, the fMOST-VTA dataset was extracted from the Ventral Tegmental Area (VTA) of a mouse brain imaged by fMOST with the size of  $9216 \times 14336 \times 24576$  voxels and has a spatial resolution of  $0.325 \times 0.325 \times 1 \mu m^3$ . We pre-processed it into the 16-bit gray image stacks with the size of  $512 \times 512 \times 512$  voxels. With the aid of neuTube open-source software (Feng, Zhao, and Kim 2015), a semi-automatic neuron reconstruction software, 25 image stacks were manually annotated by experts to serve as the ground truth. Among them, 20 were used for training and 5 for testing.

## Implementation Details

During the training process, all the 2D images are split into small patches of size  $64 \times 64$  with the batch size of 64, and all the 3D images are split into small patches of size  $16 \times 64 \times 64$  with the batch size of 16. The hyper-parameters in Eq. (11) and Eq. (14) are set to  $\lambda_d = 1$ ,  $\lambda_r = 100$ ,  $\lambda_c = 1$ , and  $\lambda_b = 1$ . The weights of binary cross entropy loss function Eq. (12) and Eq. (13) are set to  $w_c = w_b = 0.9$ . The thresholds in the MFT strategy are set to  $T_c = 0.5$  and  $T_b = 0.5$ . During the training process, the network is optimized by minimizing the centerline loss  $\lambda_c$  and the boundary loss  $\lambda_b$ . After the network converges, we freeze the parameters of U-shaped backbone, centerline block, and boundary block, and optimize the other two direction and radius blocks by minimizing the loss  $\lambda_d$  and  $\lambda_r$ . The network architecture is implemented in PyTorch 1.7.0 and MindSpore 1.7.0 (Huawei Technologies Co. 2022). The hardware configuration has 64 GB of memory, and the CPU is an Intel Xeon E5-2680 v3 and four NVIDIA GeForce RTX2080ti GPUs. With this server configuration, we took about 12

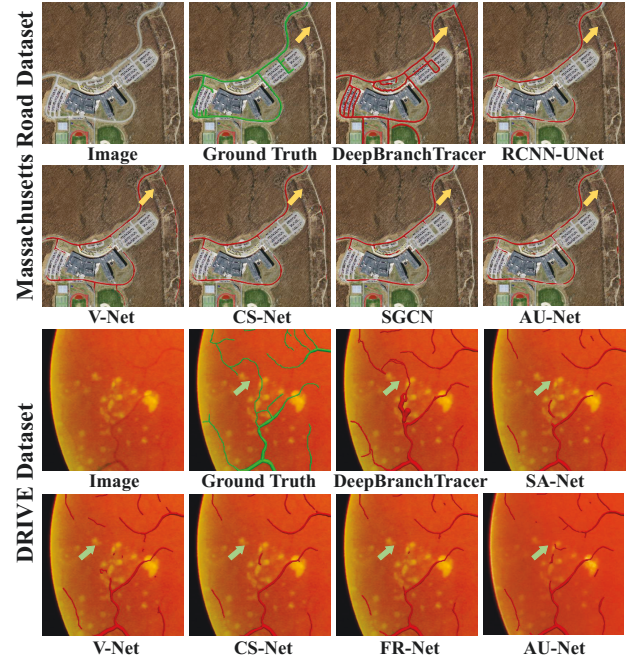


Figure 5: Reconstruction results of road and vessel datasets. The yellow arrows indicate our model detects the road which is *missing* from the ground truth label. The green arrows indicate the branches are always *broken* here.

hours to train the model. The code of this work is available at <https://github.com/CSDLLab/DeepBranchTracer>.

## Evaluation Metrics

Curvilinear structure is an elongated object, so the evaluation metric of curvilinear structure extraction should be concerned with the accuracy of its geometry rather than the classical segmentation effect. Here, we chose seven topology metrics to assess the performance of reconstruction methods. SSD-F1 metric is proposed in (Peng et al. 2010), aiming to measure how many nodes are matched in the reconstructed branch and the ground truth branch. The position error (PE) and radius error metric (RE) are Euclidean distances that calculate the position and radius of the matching node between two branches. Length metric (Wang et al. 2011; Zhang et al. 2022) (including Precision, Recall and F1-score) can be viewed as a variant of the SSD-F1 metric in terms of it tries to measure how many edges are matched. Precision indicates how accurately the predicted branch locations are. Recall captures the overall proportion of the ground truth that is effectively covered by prediction branches. We also calculated the average length of the branches (ABL), hoping that there will be more long and intact branches in the reconstruction results.

## Ablation Study

We analyzed the effects of different model settings on the fMOST-VTA dataset, as shown in Table 1. M1 only uses geometric features to estimate the direction and the radius

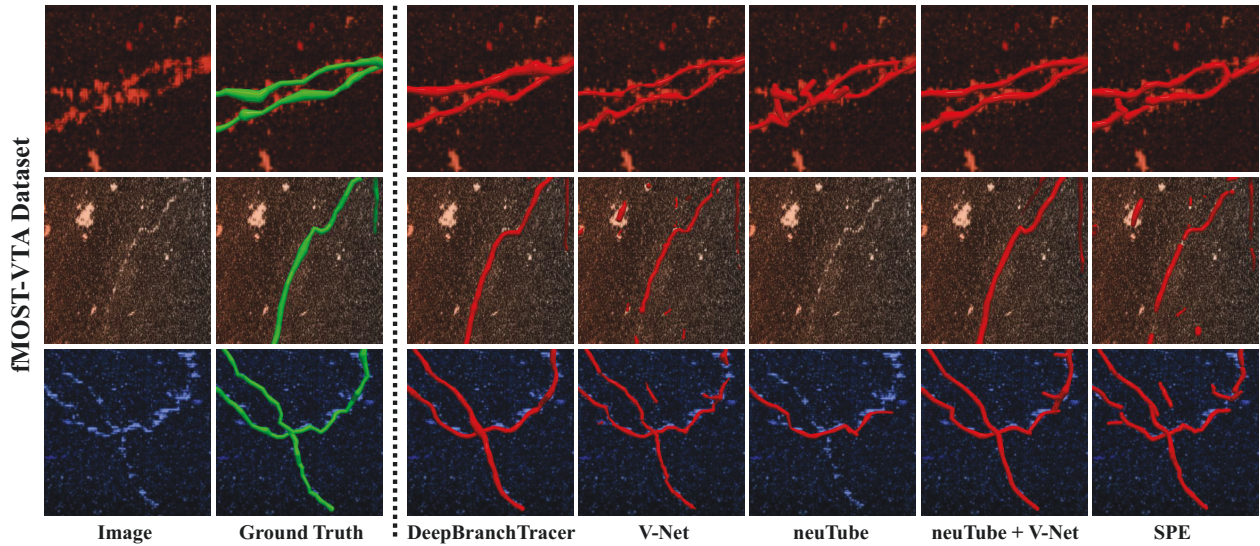


Figure 6: Reconstruction results of 3D neuron datasets. From top to bottom, we test three challenging scenarios for neuron tracing, including *adjacent* branches, *weak* branch, and *crossover* branches.

	Model	SSD-F1	Precision	Recall	Length-F1	ABL
Roads Dataset	U-Net	0.691	0.898	0.711	0.794	147.6
	V-Net	<b>0.703</b>	0.888	0.733	0.803	148.0
	AU-Net	0.700	0.899	0.723	0.801	150.4
	CS-Net	0.680	<b>0.901</b>	0.700	0.788	142.9
	RCNN-UNet	0.681	0.864	0.710	0.780	134.3
	SGCN	0.674	0.872	0.709	0.782	140.4
	RoadTracer	0.313	0.559	0.510	0.534	56.6
	Vec-road	0.635	0.767	0.805	0.786	99.8
	DBT	0.680	0.819	<b>0.812</b>	<b>0.816</b>	<b>200.2</b>

Table 2: Quantitative reconstruction results for road dataset. RoadTracer, Vec-road and DeepBranchTracer (DBT) are reconstruction-based models, while others are segmentation-based models.

of the branch. Not surprisingly, its SSD-F1 and Length-F1 was lower than the other method. M3 only uses the center-line feature to trace the branch and can be categorized as a segmentation-based method. This model yielded a lower PE score than the M1 and M2, indicating higher branch accuracy, which is also the advantage of the segmentation-based approaches. The Length-F1 score of M4 is similar to that of M3, but the ABL score is better than M3, indicating that the combination of geometric features and image features can improve the continuity of branches. It is worth noting that the ABL scores of M5 and M2 are higher than those of M4 and M1 respectively, demonstrating the effectiveness of sequence learning module. The best performance from M5 was achieved after combing all the modules and it outperforms other models in all metrics, as shown in Fig. 4.

### Comparison with the SOTA Methods

We tested our proposed model on the 2D road, vessel, and 3D neuron datasets respectively to show the performance of

	Model	SSD-F1	Precision	Recall	Length-F1	ABL
DRIVE	U-Net	0.841	0.870	0.768	0.816	44.5
	V-Net	0.847	0.870	0.773	0.819	44.4
	AU-Net	0.847	0.865	0.772	0.815	44.6
	CS-Net	0.847	0.877	0.774	0.822	44.0
	FR-Net	0.857	<b>0.885</b>	0.784	<b>0.831</b>	45.4
	SA-Net	0.856	0.874	0.784	0.827	44.3
	DBT	<b>0.893</b>	0.827	<b>0.829</b>	0.828	<b>67.7</b>
CHASE_DB1	U-Net	0.774	0.921	0.757	0.831	92.6
	V-Net	0.772	0.909	0.753	0.824	78.1
	AU-Net	0.768	<b>0.916</b>	0.751	0.825	86.1
	CS-Net	0.772	0.912	0.757	0.827	80.6
	FR-Net	0.776	0.897	0.770	0.829	90.7
	SA-Net	0.776	0.879	0.787	0.831	83.8
	DBT	<b>0.836</b>	0.827	<b>0.863</b>	<b>0.845</b>	<b>123.4</b>

Table 3: Quantitative reconstruction results for DIRVE and CHASE\_DB1 vessel datasets. DeepBranchTracer (DBT) is the only reconstruction-based models.

our method on both 2D and 3D fields.

**SOTA Baselines** In this section, we set up several comparative experiments with several segmentation-based and reconstruction-based methods. We chose U-Net, V-Net (Milletari, Navab, and Ahmadi 2016), and AU-Net (Schlemper et al. 2019) as classical deep segmentation-based methods, which can be conveniently applied to 2D and 3D data sets by replacing 2D convolution layer with 3D convolution layer. In particular, we also selected the SOTA methods designed specifically for various types of datasets for comparison. For the road dataset, we chose Vec-road (Tan et al. 2020), RoadTracer, RCNN-UNet (Yang et al. 2019) and SGCN as the comparison methods. Among them, RoadTracer is the only reconstruction-based methods. For vessel datasets, three other well-known vessel segmentation meth-

	Model	SSD-F1	Precision	Recall	Length-F1	ABL
fMOST-VTA	V-Net	0.883	0.868	0.868	0.868	71.1
	neuTube	0.680	0.833	0.552	0.664	103.8
	neuTube+V-Net	0.901	0.855	<b>0.918</b>	0.885	<b>216.3</b>
	MOST	0.385	0.429	0.255	0.320	12.6
	MOST+V-Net	0.745	0.684	0.779	0.728	23.9
	SPE	0.759	0.726	0.893	0.801	76.0
	DBT	<b>0.910</b>	<b>0.913</b>	0.909	<b>0.911</b>	198.3
DIADEM-CCF	V-net	0.335	0.596	0.550	0.572	58.5
	neuTube	0.315	0.519	0.424	0.467	63.2
	neuTube+V-Net	0.348	0.556	0.602	0.578	83.4
	MOST	0.209	0.411	0.218	0.285	25.6
	MOST+V-Net	0.274	0.509	0.339	0.407	16.3
	SPE	0.379	0.559	0.624	0.590	69.4
	DBT	<b>0.421</b>	<b>0.638</b>	<b>0.627</b>	<b>0.632</b>	<b>95.7</b>

Table 4: Quantitative reconstruction results for fMOST-VTA and DIADEM-CCF neuron datasets. neuTube+V-Net and MOST+V-Net are two reconstruction methods that rely on signal-enhanced images generated by the V-Net model.

ods were adopted as the baselines, which are CS-Net (Mou et al. 2021), FR-Net (Liu et al. 2022b) and SA-Net (Guo et al. 2021). For 3D neuron images, we also chose neuTube (Zhao et al. 2011), MOST (Ming et al. 2013), and SPE as the comparison methods. Specifically, for neuTube and MOST, which are designed with hand craft features, we employ the segmentation results of 3D V-net to enhance the neuron fiber images, which is how deep learning is currently being applied to neuron reconstruction problems, namely neuTube+V-Net and MOST+V-Net.

**Road Datasets** Table 2 summarizes the evaluation metrics for the reconstruction results of Massachusetts Roads Dataset. Among these models, our proposed model achieved the highest Length-F1 of 0.816 and Recall of 0.812. This demonstrates that our model extracts more roads compared to other methods, while still ensuring a satisfactory overall reconstruction result. The highest ABL score achieved by our model also indicates the improved continuity of the results. Most interestingly, our model also detects the road which is **missing** from the ground truth label, suggesting the high sensitivity of our method to curvilinear structures, as shown in the second rows of Fig. 5. Compared with the ground truth, these newly detected roads are mistaken for noise, resulting our method has low Precision scores.

**Vessel Datasets** The DRIVE and CHASE\_DB1 datasets are typically considered as segmentation tasks. In this study, we treat vessel segmentation as a reconstruction task, aiming to explore a novel perspective for addressing the challenge of extracting weak vessels. The results of reconstruction on the vessel datasets are shown in Table 3. Our proposed model obtained the highest SSD-F1, Recall, and ABL scores on DRIVE, and the highest SSD-F1, Recall, Length-F1 and ABL on CHASE\_DB1. It is worth noting that the Precision score of our model is low and the Recall score is the highest. We consider that the model further improves the extraction ability of weak vessels while inevitably increasing the tolerance of false positives. As shown in Fig. 5, our

model extracted more weak vessels than the other methods, and also achieved the best reconstruction effects. What’s more, with similar coverage rate of branches, our model can trace longer branches, reflected in the higher ABL scores.

**Neuron Datasets** The 3D neuron reconstruction results of the compared method on the fMOST-VTA and DIADEM-CCF datasets are shown in Table 4. It can be observed that in the fMOST-VTA image, the branch signal is always discontinuous in the microscope image due to uneven staining and optical system fluctuation, and can pose a challenge to automatic reconstruction, as shown in Fig. 6. Among all the methods, our DeepBranchTracer achieved the superior performances, including the highest SSD-F1, Precision, Length-F1 in fMOST-VTA dataset, and the highest all metrics in DIADEM-CCF dataset. The neuTube+V-Net also achieved satisfactory ABL score in fMOST-VTA dataset. However, neuTube was designed by hand-craft features and only suitable for dark field images, limiting its applicability within certain applications. The visualization results show that in most challenging scenarios, DeepBranchTracer consistently achieved better performance, demonstrating its robustness and generalization.

## Conclusion

Curvilinear structure extraction is an essential research area, but there are still challenges due to the complexity of its geometry. In this paper, we propose a novel curvilinear structure reconstruction method, namely DeepBranchTracer, which can be conveniently applied to various 2D and 3D image datasets. Specifically, our DeepBranchTracer utilizes both the external image features and internal geometric features of the curvilinear structure to finish the tracing process iteratively. Compared with the existing tracing method, our DeepBranchTracer does not require any parametric geometry model or complex training strategy, demonstrating its strong effectiveness and applicability. We evaluate our method on five road, vessel and neuron datasets, and the results show our model has a great potential in complex-topology curvilinear structure extraction.

## Acknowledgments

This work is supported by the National Key R&D Program of China (2020YFB1313500), National Natural Science Foundation of China (T2293723, 61972347), Zhejiang Provincial Natural Science Foundation (Z24F020008), the Key R&D Program of Zhejiang Province (2022C01022, 2022C01119, 2021C03003), and the Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

## References

- Bastani, F.; He, S.; Abbar, S.; Alizadeh, M.; Balakrishnan, H.; Chawla, S.; Madden, S.; and DeWitt, D. 2018. Road-tracer: Automatic extraction of road networks from aerial images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4720–4728.
- Brown, K. M.; Barrionuevo, G.; Canty, A. J.; De Paola, V.; Hirsch, J. A.; Jefferis, G. S.; Lu, J.; Snippe, M.; Sugihara,



- I.; and Ascoli, G. A. 2011. The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9: 143–157.
- Carballal, A.; Novoa, F. J.; Fernandez-Lozano, C.; García-Guimaraes, M.; Aldama-López, G.; Calviño-Santos, R.; Vazquez-Rodriguez, J. M.; and Pazos, A. 2018. Automatic multiscale vascular image segmentation algorithm for coronary angiography. *Biomedical Signal Processing and Control*, 46: 1–9.
- Chen, W.; Liu, M.; Du, H.; Radojević, M.; Wang, Y.; and Meijering, E. 2021. Deep-learning-based automated neuron reconstruction from 3D microscopy images using synthetic training images. *IEEE Transactions on Medical Imaging*, 41(5): 1031–1042.
- Chen, Z.; Deng, L.; Luo, Y.; Li, D.; Junior, J. M.; Gonçalves, W. N.; Nurunnabi, A. A. M.; Li, J.; Wang, C.; and Li, D. 2022. Road extraction in remote sensing data: A survey. *International Journal of Applied Earth Observation and Geoinformation*, 112: 102833.
- Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 764–773.
- Feng, L.; Zhao, T.; and Kim, J. 2015. neuTube 1.0: a new design for efficient neuron reconstruction software based on the SWC format. *eneuro*, 2(1).
- Gong, H.; Zeng, S.; Yan, C.; Lv, X.; Yang, Z.; Xu, T.; Feng, Z.; Ding, W.; Qi, X.; Li, A.; et al. 2013. Continuously tracing brain-wide long-distance axonal projections in mice at a one-micron voxel resolution. *Neuroimage*, 74: 87–98.
- Guo, C.; Szemenyei, M.; Yi, Y.; Wang, W.; Chen, B.; and Fan, C. 2021. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In *2020 25th international conference on pattern recognition (ICPR)*, 1236–1242. IEEE.
- Huawei Technologies Co., L. 2022. Deep Learning Frameworks. In *Artificial Intelligence Technology*, 123–135. Springer.
- Li, A.; Gong, H.; Zhang, B.; Wang, Q.; Yan, C.; Wu, J.; Liu, Q.; Zeng, S.; and Luo, Q. 2010. Micro-optical sectioning tomography to obtain a high-resolution atlas of the mouse brain. *Science*, 330(6009): 1404–1408.
- Li, R.; Zeng, T.; Peng, H.; and Ji, S. 2017. Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction. *IEEE transactions on medical imaging*, 36(7): 1533–1541.
- Liu, C.; Wang, D.; Zhang, H.; Wu, W.; Sun, W.; Zhao, T.; and Zheng, N. 2022a. Using simulated training data of voxel-level generative models to improve 3D neuron reconstruction. *IEEE transactions on medical imaging*, 41(12): 3624–3635.
- Liu, W.; Yang, H.; Tian, T.; Cao, Z.; Pan, X.; Xu, W.; Jin, Y.; and Gao, F. 2022b. Full-resolution network and dual-threshold iteration for retinal vessel and coronary angiograph segmentation. *IEEE Journal of Biomedical and Health Informatics*, 26(9): 4623–4634.
- Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.
- Meijering, E. 2010. Neuron tracing in perspective. *Cytometry Part A*, 77(7): 693–704.
- Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. Ieee.
- Ming, X.; Li, A.; Wu, J.; Yan, C.; Ding, W.; Gong, H.; Zeng, S.; and Liu, Q. 2013. Rapid reconstruction of 3D neuronal morphology from light microscopy images with augmented rayburst sampling. *PloS one*, 8(12): e84557.
- Mnih, V. 2013. *Machine learning for aerial image labeling*. University of Toronto (Canada).
- Mosinska, A.; Koziński, M.; and Fua, P. 2019. Joint segmentation and path classification of curvilinear structures. *IEEE transactions on pattern analysis and machine intelligence*, 42(6): 1515–1521.
- Mou, L.; Zhao, Y.; Fu, H.; Liu, Y.; Cheng, J.; Zheng, Y.; Su, P.; Yang, J.; Chen, L.; Frangi, A. F.; et al. 2021. CS2-Net: Deep learning segmentation of curvilinear structures in medical imaging. *Medical image analysis*, 67: 101874.
- Peng, H.; Long, F.; Zhao, T.; and Myers, E. 2011. Proof-editing is the bottleneck of 3D neuron reconstruction: the problem and solutions. *Neuroinformatics*, 9: 103–105.
- Peng, H.; Ruan, Z.; Long, F.; Simpson, J. H.; and Myers, E. W. 2010. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28(4): 348–353.
- Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; and Yang, G. 2023. Dynamic Snake Convolution based on Topological Geometric Constraints for Tubular Structure Segmentation. *arXiv preprint arXiv:2307.08388*.
- Rodriguez, A.; Ehlenberger, D. B.; Hof, P. R.; and Wearne, S. L. 2009. Three-dimensional neuron tracing by voxel scooping. *Journal of neuroscience methods*, 184(1): 169–175.
- Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, 234–241. Springer.
- Schlemper, J.; Oktay, O.; Schaap, M.; Heinrich, M.; Kainz, B.; Glocker, B.; and Rueckert, D. 2019. Attention gated networks: Learning to leverage salient regions in medical images. *Medical image analysis*, 53: 197–207.
- Soomro, T. A.; Afifi, A. J.; Zheng, L.; Soomro, S.; Gao, J.; Hellwich, O.; and Paul, M. 2019. Deep learning models for retinal blood vessels segmentation: a review. *IEEE Access*, 7: 71696–71717.
- Staal, J.; Abramoff, M. D.; Niemeijer, M.; Viergever, M. A.; and Van Ginneken, B. 2004. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4): 501–509.

- Tan, Y.-Q.; Gao, S.-H.; Li, X.-Y.; Cheng, M.-M.; and Ren, B. 2020. Vecroad: Point-based iterative graph exploration for road graphs extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8910–8918.
- Türetken, E.; Benmansour, F.; Andres, B.; Glowacki, P.; Pfister, H.; and Fua, P. 2016. Reconstructing curvilinear networks using path classifiers and integer programming. *IEEE transactions on pattern analysis and machine intelligence*, 38(12): 2515–2530.
- Wang, Y.; Narayanaswamy, A.; Tsai, C.-L.; and Roysam, B. 2011. A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9: 193–217.
- Yang, B.; Liu, M.; Wang, Y.; Zhang, K.; and Meijering, E. 2021. Structure-guided segmentation for 3D neuron reconstruction. *IEEE transactions on medical imaging*, 41(4): 903–914.
- Yang, X.; Li, X.; Ye, Y.; Lau, R. Y.; Zhang, X.; and Huang, X. 2019. Road detection and centerline extraction via deep recurrent convolutional neural network U-Net. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9): 7209–7220.
- Yang, X.; Li, Z.; Guo, Y.; and Zhou, D. 2022. DCU-net: A deformable convolutional neural network based on cascade U-net for retinal vessel segmentation. *Multimedia Tools and Applications*, 81(11): 15593–15607.
- Yuan, X.; Trachtenberg, J. T.; Potter, S. M.; and Roysam, B. 2009. MDL constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7: 213–232.
- Zhang, H.; Liu, C.; Yu, Y.; Dai, J.; Zhao, T.; and Zheng, N. 2022. PyNeval: A Python Toolbox for Evaluating Neuron Reconstruction Performance. *Frontiers in Neuroinformatics*, 15: 767936.
- Zhao, T.; Xie, J.; Amat, F.; Clack, N.; Ahammad, P.; Peng, H.; Long, F.; and Myers, E. 2011. Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9: 247–261.
- Zhou, G.; Chen, W.; Gui, Q.; Li, X.; and Wang, L. 2021. Split depth-wise separable graph-convolution network for road extraction in complex environments from high-resolution remote-sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–15.