

# CAPE: Connectivity-Aware Path Enforcement Loss for Curvilinear Structure Delineation

Elyar Esmailzadeh<sup>[0009–0003–5610–2604]</sup>, Ehsan Garaaghaji<sup>[0009–0003–2720–7415]</sup>, Farzad Hallaji Azad<sup>[0009–0003–3882–4275]</sup>, and Doruk Oner <sup>\*</sup><sup>[0000–0002–9403–4628]</sup>

NeuraVision Lab, Department of Computer Engineering,  
Bilkent University, Ankara, Turkey  
{elyar,ehsan.garaaghaji,farzad.hallaji,doruk.oner}@bilkent.edu.tr

**Abstract.** Promoting the connectivity of curvilinear structures, such as neuronal processes in biomedical scans and blood vessels in CT images, remains a key challenge in semantic segmentation. Traditional pixel-wise loss functions, including cross-entropy and Dice losses, often fail to capture high-level topological connectivity, resulting in topological mistakes in graphs obtained from prediction maps. In this paper, we propose CAPE (Connectivity-Aware Path Enforcement), a novel loss function designed to enforce connectivity in graphs obtained from segmentation maps by optimizing a graph connectivity metric. CAPE uses the graph representation of the ground truth to select node pairs and determine their corresponding paths within the predicted segmentation through a shortest-path algorithm. Using this, we penalize both disconnections and false positive connections, effectively promoting the model to preserve topological correctness. Experiments on 2D and 3D datasets, including neuron and blood vessel tracing demonstrate that CAPE significantly improves topology-aware metrics and outperforms state-of-the-art methods. Code is available at <https://github.com/NeuraVisionLab/CAPE>.

**Keywords:** Segmentation · Topology Optimization · Biomedical Imaging · Neuronal Process Tracing · Blood Vessel Segmentation.

## 1 Introduction

Segmenting curvilinear structures—such as neuronal processes in biomedical scans and blood vessels in CT images—is essential for numerous applications in medical imaging and biological research. Traditional deep convolutional networks have significantly improved pixel-level segmentation. However, despite these advancements, they exhibit a fundamental limitation: they are trained using pixel-wise loss functions such as cross-entropy, Dice, or Mean Squared Error (MSE), which do not explicitly enforce topological correctness, because they are designed to optimize the number of correctly segmented pixels/voxels, but do not account for high-level structural continuity. This is particularly problematic, as

---

\* Corresponding author: [doruk.oner@bilkent.edu.tr](mailto:doruk.oner@bilkent.edu.tr)

connectivity errors can severely compromise the reliability of constructions and their usability in downstream tasks. As a result, annotators are often required to manually correct topological errors, a highly labor-intensive and tedious process, especially when dealing with large volumetric datasets.

Recent approaches have focused on designing loss functions specifically aimed at enforcing topological connectivity [16,4,10,3,2,22,18,19]. Some methods analyze the structural topology using Persistent Homology [4,10,3,2], while others incorporate perceptual loss by leveraging pre-trained feature extractors [16]. Additionally, certain approaches promote connectivity by enforcing background separation through Rand index optimization [18,19]. Another strategy involves developing differentiable loss functions based on existing metrics [22]. Despite these efforts, all existing methods attempt to achieve global connectivity by optimizing a metric that is not explicitly designed for connectivity. Consequently, they only indirectly enforce the global topological correctness of predictions.

In this paper, we propose CAPE (Connectivity-Aware Path Enforcement), a novel loss function that improves the connectivity of curvilinear structure reconstructions. Inspired by the Average Path Length Similarity (APLS) metric [5], CAPE evaluates connectivity by comparing entire paths instead of individual pixels. Specifically, it leverages Dijkstra’s algorithm to compute shortest paths in the pixel domain by mapping paths between nodes from the ground truth graph onto the predicted segmentation. This direct comparison enforces high-level topological correctness, leading to improved segmentation.

## 2 Related Work

Over the years, various approaches have been proposed for delineating curvilinear structures, ranging from hand-designed filters [6,13,26] to learned filters [30,1] using support vector machines [11], gradient boosting [24], and decision trees [27]. However, these traditional methods suffer from limited generalizability and various shortcomings. With the advancements in deep learning, more powerful deep neural networks have largely replaced these techniques [15,8,14,20,9,29].

Recent advances in biomedical image segmentation have been driven by deep convolutional neural networks, with U-Net [21] being especially influential. While U-Net achieves high pixel-level accuracy, conventional pixel-wise loss functions neglect topological correctness—a critical issue for curvilinear structures like neuronal processes, where connectivity errors degrade segmentation quality. Consequently, many studies have explored methods beyond pixel-wise losses to fully harness the potential of deep networks.

Several studies have designed novel topology-aware loss functions to improve segmentation performance of deep networks. [16] penalized differences in label and prediction feature maps using a pretrained VGG19 network [23], but it lacks explicit topological preservation. Persistent Homology-based methods [4,10,3] provide differentiable topological similarity but ignore spatial alignment, leading to inaccuracies. [17] introduced a filtration function addressing this issue, though persistent homology remains computationally expensive. The soft-clDice

loss [22] enhances connectivity via soft-skeletonization but remains pixel-wise and volumetric-focused. MALIS loss [7] learns region affinities but struggles with curvilinear structures, because of the existence of loops in those structures. [18] proposed inverse MALIS for curvilinear connectivity by enforcing regional separation, effective in 2D but not generalizing to 3D. To address this, [19] introduced a projection-based extension for 3D, yet it relies on Rand Index, which is suboptimal for curvilinear structures and yields sparse gradients, limiting global connectivity enforcement. Additionally, it often penalizes a single pixel per background region, resulting in highly sparse gradients, making the training less stable.

To evaluate networks for curvilinear structure segmentation, we use metrics beyond the Rand Index, such as *Average Path Length Similarity* (APLS) [5]. APLS, a graph-based metric, quantifies similarity by comparing shortest path distances between corresponding nodes, penalizing deviations in length and structure to assess connectivity. Unlike pixel-wise metrics, connectivity-based metrics evaluate groups of pixels or voxels, making conventional metrics ineffective for topological correctness. However, using APLS and TLTS [28] as loss functions is impractical due to their non-differentiability in the graph domain.

While existing methods achieve competitive results, none directly optimize connectivity-based metrics. CAPE, inspired by APLS, introduces a differentiable, pixel-domain variant. Unlike prior methods [18,19], which yield sparse gradients penalizing individual pixels, CAPE produces denser gradients along the entire path, enhancing connectivity enforcement while remaining suitable for gradient-based optimization.

### 3 Methodology

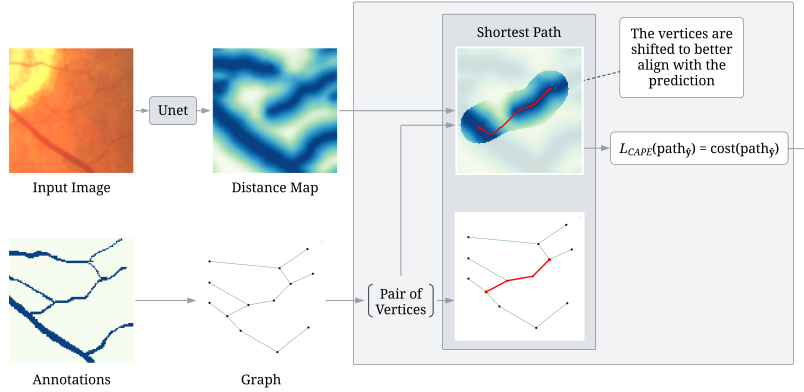
Our approach is inspired by the APLS connectivity metric, which evaluates the connectivity of curvilinear structures but is inherently non-differentiable and unsuitable for direct integration into training segmentation networks. In APLS, edges are binary—either present or absent. In our differentiable adaptation, we replace these binary affinities with continuous ones (distance values predicted by our network). This allows path lengths to be computed as sums of network predictions, enabling optimization. At a high level, our loss function measures the consistency of connectivity between the ground truth and the network’s prediction by comparing the costs of corresponding shortest paths. Specifically, we consider the ground truth graph  $G = (V, E)$ , where vertices  $V$  represent key structural points and edges  $E$  capture the true connections among foreground elements. Our network,  $\mathbf{f}_\theta$ , predicts a distance map  $\hat{\mathbf{y}}$  in which each pixel indicates its distance from the nearest foreground structure (for brevity, we refer to image elements as pixels; in 3D data, these correspond to voxels, and the same methodology applies). We predict distance maps instead of probability maps because distance map pixel values rise linearly as disconnections increase, more effectively penalizing longer or severe gaps. Probability maps, however, saturate near zero in disconnected regions, reducing loss sensitivity.

### 3.1 CAPE Loss

In an iterative procedure, we randomly select two connected vertices  $v_1$  and  $v_2$  from  $G$  and compute their ground truth shortest path,  $\text{path}_G$ , using Dijkstra’s algorithm. We then project these vertices onto the predicted distance map by selecting the local minima within a constrained  $7 \times 7$  neighborhood to account for small deviations between  $G$  and the prediction centerlines, yielding refined positions  $v'_1$  and  $v'_2$ . A second Dijkstra pass on the predicted distance map computes the predicted shortest path,  $\text{path}_{\hat{y}}$ . The cost of  $\text{path}_{\hat{y}}$  is defined as,

$$\text{cost}(\text{path}_{\hat{y}}) = \sum_{n \in \text{path}_{\hat{y}}} \hat{y}(n)^2, \quad (1)$$

where  $n$  is the set of pixels in  $\text{path}_{\hat{y}}$ , and  $\hat{y}(n)$  is the predicted distance value for pixel  $n$ . We use the square of the distance predictions to ensure that pixels with higher distance values—indicating more severe disconnections—contribute more significantly to the overall loss, thereby generating stronger gradients compared to those with mild disconnections. This cost is fully differentiable with respect to the network parameters, making our loss function well-suited for gradient-based optimization.



**Fig. 1. Computation of  $L_{CAPE}$ .** After extracting the ground truth graph, an iterative process selects pairs of vertices and computes their shortest path. The corresponding path is then masked with dilation and then projected to the pixel domain, and the shortest path algorithm is reapplied to obtain  $L_{CAPE}$ .

For a corresponding pair of paths from the ground truth,  $\text{path}_G$ , and prediction,  $\text{path}_{\hat{y}}$ , the loss is defined as the absolute difference between their costs. Because the ground truth graph embodies perfect connectivity, we define the cost of any ground truth path  $\text{path}_G$  as 0, which simplifies the loss to cost of  $\text{path}_{\hat{y}}$

$$L_{CAPE}(\text{path}_G, \text{path}_{\hat{y}}) = |\text{cost}(\text{path}_G) - \text{cost}(\text{path}_{\hat{y}})| = \text{cost}(\text{path}_{\hat{y}}). \quad (2)$$

If the network confidently captures the connection, the cost along the predicted path will be close to zero, yielding a low loss; conversely, if there is a disconnection, high distance values along the path will increase the cost and result in a higher loss.

The iterative procedure continues until all connections in  $G$  are evaluated. After processing each pair, the corresponding edges are removed from  $G$  to ensure unique processing, and the procedure repeats until no edges remain. The complete procedure for computing the loss can be found in Algorithm 1.

---

**Algorithm 1** Computation of  $L_{\text{CAPE}}$ 


---

**Input:** Groundtruth Graph  $G = (V, E)$  and predicted distance map  $\hat{\mathbf{y}}$

**Output:**  $L_{\text{CAPE}}(G, \hat{\mathbf{y}})$

```

1: Initialize  $L_{\text{CAPE}} \leftarrow 0$ 
2: while  $E \neq \emptyset$  do
3:   Pick a pair of vertices,  $(v_1, v_2)$ , that are connected from  $G$ 
4:    $\text{path}_G \leftarrow \text{Dijkstra}(G, v_1, v_2)$ 
5:    $v'_1, v'_2 \leftarrow \text{shift}(v_1), \text{shift}(v_2)$ 
6:    $M \leftarrow \text{render}(\text{path}_G)$ 
7:    $M_{\text{dilated}} \leftarrow \text{dilate}(M)$ 
8:    $\text{path}_{\hat{\mathbf{y}}} \leftarrow \text{Dijkstra}(\hat{\mathbf{y}} \cdot M_{\text{dilated}}, v'_1, v'_2)$ 
9:    $L_{\text{CAPE}} \leftarrow L_{\text{CAPE}} + \text{cost}(\text{path}_{\hat{\mathbf{y}}})$ 
10:  for  $e \in \text{path}_G$  do
11:    Remove  $e$  from  $E$ 
12:  end for
13: end while
14: return  $L_{\text{CAPE}}$ 

```

---

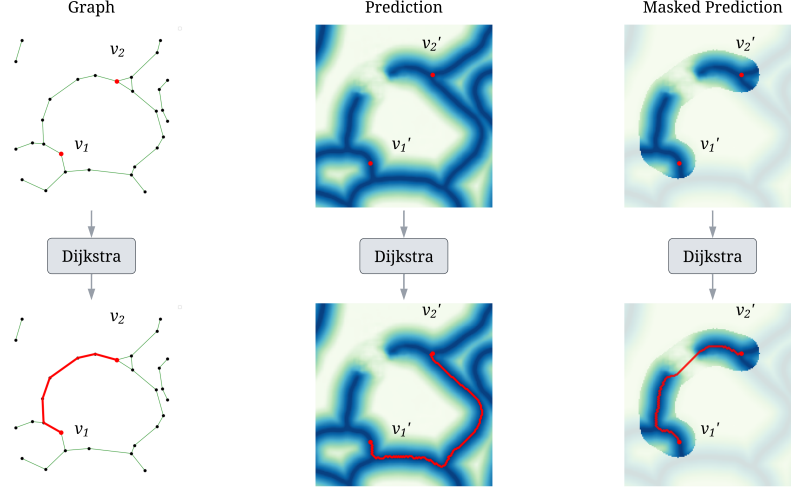
The reason we perform a second Dijkstra on the predicted distance map rather than directly using the ground truth shortest path for cost computation is to allow for minor spatial deviations in the predicted path. Centerline annotations can be noisy, and enforcing exact spatial correspondence would undesirably penalize predictions that correctly capture connectivity despite small shifts. By computing the predicted path cost via a separate Dijkstra pass on  $\hat{\mathbf{y}}$ , our loss function focuses on evaluating connectivity rather than strict spatial alignment. The computation of loss is explained visually in Figure 1.

In addition, we incorporate a masking strategy to further constrain the search for the predicted path. First, we render the ground truth path to obtain a binary mask  $M$ , then dilate this mask by 10 pixels to create a dilated mask  $M_{\text{dilated}}$ .

The dilated mask is applied element-wise to the predicted distance map  $\hat{\mathbf{y}}$ , and the shortest path is computed within the masked region between the projected vertices  $v'_1$  and  $v'_2$  using Dijkstra’s algorithm:

$$\text{path}_{\hat{\mathbf{y}}} = \text{Dijkstra}(\hat{\mathbf{y}} \cdot M_{\text{dilated}}, v'_1, v'_2). \quad (3)$$

This approach confines the search to a region closely aligned with the ground truth, allowing only limited deviations and reducing the likelihood of selecting



**Fig. 2. Masking Strategy.** Left: Ground truth graph with selected vertices and the shortest path computed via Dijkstra’s algorithm. Middle: Projected vertices on the predicted distance map and the corresponding Dijkstra path computed without masking; due to a loop, the algorithm bypasses the disconnection. Right: With a mask applied around the ground truth path, Dijkstra is forced to capture the disconnection by preventing an alternate loop path.

an alternative, longer path—especially in cases where loops are present—while also lowering the computational complexity of the loss calculation. As illustrated in Figure 2, masking effectively guides the selection of matching paths, ensuring that the intended shortest path is chosen even in loopy structures.

### 3.2 Training Loss Integration

The total CAPE loss,  $L_{\text{CAPE}}(G, \hat{\mathbf{y}})$ , is defined as the sum of the losses computed for each ground truth path sampled from the graph  $G$ .

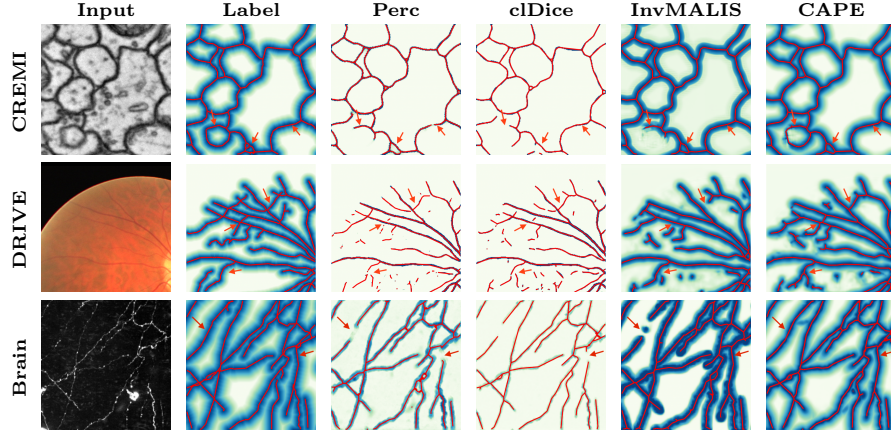
$$L_{\text{CAPE}}(G, \hat{\mathbf{y}}) = \sum_{\text{path}_G \in \mathcal{P}_G} L_{\text{CAPE}}(\text{path}_G, \text{path}_{\hat{\mathbf{y}}}) = \sum_{\text{path}_G \in \mathcal{P}_G} \text{cost}(\text{path}_{\hat{\mathbf{y}}}), \quad (4)$$

where  $\mathcal{P}_G$  is the set of paths sampled from  $G$ .

In our training framework, we combine the CAPE loss with a per-pixel Mean Squared Error (MSE) loss to provide comprehensive supervision. While the CAPE loss delivers connectivity-aware gradients primarily along foreground paths, the MSE loss ensures dense supervision across all pixels, including background regions. The total loss function is defined as

$$L_{\text{TOTAL}} = L_{\text{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) + \alpha L_{\text{CAPE}}(G, \hat{\mathbf{y}}), \quad (5)$$

where  $\mathbf{y}$  is the ground truth per-pixel distance map computed from the graph  $G$  and  $\alpha$  is a hyperparameter to balance the per-pixel and connectivity-aware components of the loss.



**Fig. 3.** Qualitative comparison of the test results in 2D and 3D datasets. The arrows highlight regions that remain disconnected in alternative methods but are successfully connected using CAPE. The connectivity improves significantly when our approach is used.

## 4 Experiments

### 4.1 Datasets

We conduct experiments on 2D and 3D datasets. For 2D experiments, we use two datasets: **CREMI** [7], which contains 83 training and 42 validation samples of size  $1250 \times 1250$ , including neurons from the *Drosophila melanogaster* brain captured by serial section electron microscopy, and the **DRIVE** dataset [25], which contains 13 training and 7 validation samples from digital retinal images of blood vessels, sized  $584 \times 565$ . For the 3D experiments, we use the **Brain** dataset, which consists of 14 light microscopy scans of the mouse brain, each of size  $250 \times 250 \times 200$ . We use 10 of them for training and 4 for validation.

### 4.2 Metrics

We evaluate our method using two pixel-wise metrics, **CCQ** and **Dice**, and two connectivity-aware metrics, **APLS** and **TLTS**. **CCQ** (Correctness, Completeness, and Quality) is analogous to precision, recall, and F1-score, with true positives defined as pixels within 3 units of the ground truth, making it well-suited for centerline segmentation. **Dice** score measures the overlap between predicted

**Table 1.** Comparative results. As seen in the table, our method not only gives state-of-the-art results in the topology-aware metrics, but also comparative results in pixel-wise metrics.

Dataset	Methods	Pixel-wise				Topology-aware	
		Corr.	Comp.	Qual.	Dice	APLS	TLTS
CREMI	MSE	<u>98.3</u>	99.6	97.9	85.4	81.9	79.5
	Perc	<b>99.9</b>	98.8	<b>98.7</b>	<b>88.9</b>	91.6	91.2
	clDice	99.0	96.5	95.7	79.3	82.3	81.2
	InvMALIS	98.2	<u>99.7</u>	97.9	87.9	<u>91.9</u>	<u>92.4</u>
	OURS	<u>98.3</u>	<b>99.8</b>	<u>98.1</u>	<u>88.7</u>	<b>93.6</b>	<b>93.5</b>
DRIVE	MSE	<b>97.2</b>	92.6	<u>90.4</u>	77.1	68.2	66.9
	Perc	<u>96.3</u>	93.5	<u>90.4</u>	77.3	76.8	72.8
	clDice	94.2	93.8	88.8	76.2	74.5	70.5
	InvMALIS	94.6	<b>94.7</b>	89.9	<u>79.3</u>	<u>77.7</u>	<u>74.5</u>
	OURS	95.2	<u>94.6</u>	<b>90.5</b>	<b>80.3</b>	<b>80.2</b>	<b>76.4</b>
Brain	MSE	<b>99.0</b>	94.1	93.2	75.2	83.9	75.2
	Perc	97.1	<b>97.3</b>	<u>94.6</u>	<b>80.4</b>	83.5	<u>86.4</u>
	clDice	<u>98.3</u>	96.5	<b>94.9</b>	70.6	84.4	85.3
	InvMALIS	96.6	96.2	93.0	68.2	<u>85.7</u>	83.8
	OURS	97.1	<u>97.2</u>	94.4	<u>78.3</u>	<b>89.8</b>	<b>93.1</b>

and true segmentations. For connectivity, **APLS** (Average Path Length Similarity) compares the average shortest path lengths between corresponding nodes in the ground truth and predicted graphs, while **TLTS** (Too-Long-Too-Short) quantifies the fraction of predicted paths whose lengths deviate by less than 15% from the ground truth.

### 4.3 Architectures and Baselines

For our 2D experiments, we use a 2D U-Net [21], featuring three down-samplings, two convolutional layers per level, max-pooling for the encoder, and bilinear up-sampling for the decoder. For 3D experiments, we employ a 3D U-Net with the same configuration. The 2D models are trained for 10k epochs and the 3D models are trained for 50k epochs, both with Adam optimizer [12] with a learning rate of  $1e-3$  and weight decay of  $1e-3$ . We compare our results against **MSE** loss ( $L_{MSE}$ ) between predictions and ground truths, a network trained with **Perceptual loss** using a pre-trained VGG19 [16,23], **clDice** loss [22], and **InvMALIS**. For 2D datasets, we compare to inverse MALIS proposed in [18], and for 3D, we compare to the inverse MALIS computed on projections [19].

### 4.4 Results

Figure 3 and Table 1 present our qualitative and quantitative results, respectively. As shown in Table 1, our method significantly improves topology-aware



metrics and outperforms state-of-the-art approaches, while achieving comparable performance on pixel-wise metrics. The qualitative results further highlight the enhanced connectivity in our segmentations.

## 5 Conclusion and Future Work

In this paper, we introduce CAPE (Connectivity-Aware Path Enforcement), a loss function that optimizes topological correctness by comparing ground truth and predicted shortest paths via Dijkstra’s algorithm. By generating denser gradients along entire paths, CAPE effectively penalizes disconnections and promotes connectivity. Experiments on 2D and 3D datasets demonstrate significant improvements in connectivity metrics, confirming its effectiveness for curvilinear structure segmentation in biomedical imaging. In future work, we plan to have the network directly output graph representations rather than distance maps, enabling CAPE to operate entirely in the graph domain. This unified approach is expected to further enhance topological accuracy and connectivity enforcement.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Breitenreicher, D., Sofka, M., Britzen, S., Zhou, S.K.: Hierarchical discriminative framework for detecting tubular structures in 3d images. In: Information Processing in Medical Imaging. pp. 328–339 (2013)
2. Byrne, N., Clough, J.R., Montana, G., King, A.P.: A persistent homology-based topological loss function for multi-class cnn segmentation of cardiac mri. In: Statistical Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges. pp. 3–13 (2021)
3. Clough, J.R., Byrne, N., Oksuz, I., Zimmer, V.A., Schnabel, J.A., King, A.P.: A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 8766–8778 (12 2022)
4. Edelsbrunner, H., Harer, J.: Persistent homology — a survey. In: Surveys on discrete and computational geometry. vol. 453, p. 257. Amer Mathematical Society (2008)
5. Etten, A.: Spacenet road detection and routing challenge part ii—apls implementation (2017)
6. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: Medical Image Computing and Computer-Assisted Intervention. pp. 130–137 (1998)
7. Funke, J., Tschopp, F., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., Turaga, S.C.: Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence* **41**(7), 1669–1680 (2018)
8. Ganin, Y., Lempitsky, V.:  $n^4$ -fields: Neural network nearest neighbor fields for image transforms. In: Computer Vision – ACCV. pp. 536–551 (2015)

9. Guo, Z., Bai, J., Lu, Y., Wang, X., Cao, K., Song, Q., Sonka, M., Yin, Y.: Deep-centerline: A multi-task fully convolutional network for centerline extraction. In: Information Processing in Medical Imaging. pp. 441–453 (2019)
10. Hu, X., Li, F., Samaras, D., Chen, C.: Topology-preserving deep image segmentation. In: Advances in Neural Information Processing Systems. vol. 32 (2019)
11. Huang, X., Zhang, L.: Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. *International Journal of Remote Sensing* **30**(8), 1977–1987 (2009)
12. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. *International Conference on Learning Representations* (2015)
13. Law, M.W.K., Chung, A.C.S.: Three dimensional curvilinear structure detection using optimally oriented flux. In: Computer Vision – ECCV. pp. 368–382 (2008)
14. Maninis, K.K., Pont-Tuset, J., Arbeláez, P., Van Gool, L.: Deep retinal image understanding. In: Medical Image Computing and Computer-Assisted Intervention. pp. 140–148 (2016)
15. Mnih, V., Hinton, G.E.: Learning to detect roads in high-resolution aerial images. In: Computer Vision – ECCV. pp. 210–223 (2010)
16. Mosinska, A., Márquez-Neila, P., Koziński, M., Fua, P.: Beyond the pixel-wise loss for topology-aware delineation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition* pp. 3136–3145 (2017)
17. Oner, D., Garin, A., Koziński, M., Hess, K., Fua, P.: Persistent homology with improved locality information for more effective delineation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(8), 10588–10595 (2023)
18. Oner, D., Koziński, M., Citraro, L., Dadap, N.C., Konings, A.G., Fua, P.: Promoting Connectivity of Network-Like Structures by Enforcing Region Separation. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **44**(09), 5401–5413 (2022)
19. Oner, D., Osman, H., Koziński, M., Fua, P.: Enforcing connectivity of 3d linear structures using their 2d projections. In: Medical Image Computing and Computer Assisted Intervention 2022. pp. 591–601 (2022)
20. Peng, H., Zhou, Z., Meijering, E.e.a.: Automatic tracing of ultra-volumes of neuronal images. In: *Nature Methods* **14** (2017)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention 2015. pp. 234–241 (2015)
22. Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J.P., Bauer, U., Menze, B.H.: cldice-a novel topology-preserving loss function for tubular structure segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16560–16569 (2021)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations* (2015)
24. Sironi, A., Lepetit, V., Fua, P.: Multiscale centerline detection by learning a scale-space distance transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
25. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., Van Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging* **23**(4), 501–509 (2004)
26. Turetken, E., Becker, C., Glowacki, P., Benmansour, F., Fua, P.: Detecting irregular curvilinear structures in gray scale and color imagery using multi-directional oriented flux. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1553–1560. Institute of Electrical and Electronics Engineers Inc. (2013)

27. Türetken, E., Benmansour, F., Andres, B., Glowacki, P., Pfister, H., Fua, P.: Reconstructing curvilinear networks using path classifiers and integer programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(12), 2515–2530 (2016)
28. Wegner, J.D., Montoya-Zegarra, J.A., Schindler, K.: A higher-order crf model for road network extraction. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1698–1705 (2013)
29. Wolterink, J.M., van Hamersvelt, R.W., Viergever, M.A., Leiner, T., Išgum, I.: Coronary artery centerline extraction in cardiac ct angiography using a cnn-based orientation classifier. *Medical Image Analysis* **51**, 46–60 (2019)
30. Wu, D., Liu, D., Puskas, Z., Lu, C., Wimmer, A., Tietjen, C., Soza, G., Zhou, S.K.: A learning based deformable template matching method for automatic rib centerline extraction and labeling in ct images. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 980–987 (2012)