



DeepCenterline: A Multi-task Fully Convolutional Network for Centerline Extraction

Zhihui Guo¹, Junjie Bai²(✉), Yi Lu², Xin Wang², Kunlin Cao², Qi Song², Milan Sonka¹, and Youbing Yin²

¹ University of Iowa, Iowa City, IA 52242, USA

² CuraCloud Corporation, Seattle, WA 98104, USA

junjieb@curacloudcorp.com

Abstract. A novel centerline extraction framework is reported which combines an end-to-end trainable multi-task fully convolutional network (FCN) with a minimal path extractor. The FCN simultaneously computes centerline distance maps and detects branch endpoints. The method generates single-pixel-wide centerlines with no spurious branches. It handles arbitrary tree-structured object with no prior assumption regarding depth of the tree or its bifurcation pattern. It is also robust to substantial scale changes across different parts of the target object and minor imperfections of the object's segmentation mask. To the best of our knowledge, this is the first deep-learning based centerline extraction method that guarantees single-pixel-wide centerline for a complex tree-structured object.

The proposed method is validated in coronary artery centerline extraction on a dataset of 620 patients (400 of which used as test set). This application is challenging due to the large number of coronary branches, branch tortuosity, and large variations in length, thickness, shape, etc. The proposed method generates well-positioned centerlines, exhibiting lower number of missing branches and is more robust in the presence of minor imperfections of the object segmentation mask. Compared to a state-of-the-art traditional minimal path approach, our method improves patient-level success rate of centerline extraction from 54.3% to 88.8% according to independent human expert review.

Keywords: Centerline · Deep learning · Multi-task · Attention

1 Introduction

Centerline, or skeleton, provides a concise representation of the object topology. An ideal centerline extraction algorithm generates centerline points close enough to “centers” of the object cross-sectionally, captures all “branches”, and has no false positive spurious branches.

Z. Guo and J. Bai—Equal contribution.

© Springer Nature Switzerland AG 2019

A. C. S. Chung et al. (Eds.): IPMI 2019, LNCS 11492, pp. 441–453, 2019.

https://doi.org/10.1007/978-3-030-20351-1_34

Many semi-automated and automated approaches exist for centerline extraction. Morphological thinning and erosion based methods are popular in road centerline extraction [2]. However, centerlines extracted by these methods often come along with spurious branches and usually need ad-hoc post pruning.

In contrast, minimal path based centerline extraction methods guarantee a more structured centerline output by requiring explicitly specified endpoints. A *centerline distance map* (also called cost image) is generated from an object segmentation mask by methods such as Euclidean distance transform, which assigns smaller values to voxels closer to centerline and larger values to voxels farther away. A minimal path between the endpoints in the centerline distance map thus corresponds to the object centerline.

Minimal path algorithms are widely used in blood vessel centerline extraction [1, 2, 4, 5, 9]. In [4], Metz *et al.* adopted vesselness and region statistics as cost metrics and manually specified each branch endpoint. Gülsün *et al.* [1] used human selected features to compute flow field as cost image. Mirikharaji *et al.* [5] integrated a predefined tree topology and tubularity scores to get minimal paths. Zheng *et al.* [9] used a machine learning based vesselness algorithm to generate the cost image. These methods require either human crafted features/priors or manual specification of branch endpoints.

Convolutional neural networks (CNN) have been prevalent in medical image analysis recently and achieved great success. There are three advantages of CNN-based methods over traditional methods. First, multi-layer CNNs have enough capacity to learn complex functions that cannot be described by simple models. Second, CNNs do not require humans to select features and support end-to-end training. Third, a single CNN model has the ability to handle multiple tasks.

Coronary computed tomography angiography (CCTA) is a noninvasive technique widely used in clinical practice for coronary artery disease detection. Given a coronary artery segmentation mask, extracting its centerline is a prerequisite step for automatic stenosis grading, calcium evaluation, plaque evaluation and visualization [3, 7]. Extracting coronary artery centerline from a segmentation mask faces multiple notable challenges. First, multiple branches, usually more than a dozen, with large intra-subject and inter-subject variations of length, thickness, and shape are presented, forming a complex tree structure. Detecting all branches without false positive is quite challenging. Second, branch diameter changes significantly from proximal to distal portion of coronary artery. The proximal end can be several times wider than the distal end. Third, tortuous course of vessel branches hinders the performance of a minimal path based algorithm, by which straight paths are inherently preferred. Fourth, imperfections of segmentation masks such as brief touching of two nearby branches could lead to incorrect bifurcation in centerline.

To address these challenges, we propose a two-head multi-task FCN which simultaneously generates a locally normalized distance map and a list of branch endpoints (Fig. 1). One head of the multi-task FCN outputs a normalized centerline distance map that is scale-invariant and robust to image segmentation imperfections. Log-transform and attention mechanism are also incorporated to increase model sensitivity. The other head of the FCN automatically detects the

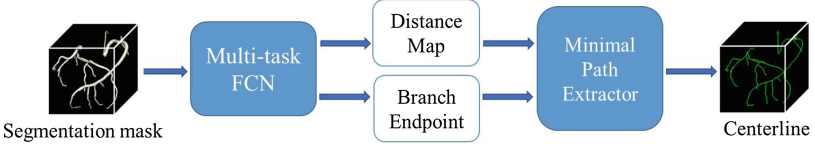


Fig. 1. Schematic workflow of DeepCenterline

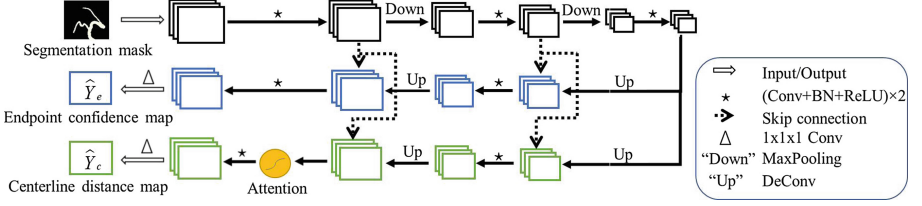


Fig. 2. The proposed multi-task FCN architecture. The input is 3D segmentation mask volume. The two tasks, centerline distance map and endpoint confidence map computation, share the same encoder path and have separate decoder paths. Skip-connections are added among features of same scale to facilitate good use of information. An attention module is added for the centerline distance map task to further boost accuracy.

sparsely distributed endpoints of the object skeleton with high accuracy. The resulting distance map and endpoint list are fed into a minimal path extractor which gives the final centerline extraction results.

2 Method

The proposed method consists of two main steps (Fig. 1): a multi-task FCN computing a locally normalized centerline distance map and a list of endpoints simultaneously, and a minimal cost path extractor taking the output of the FCN to generate a set of paths as centerline.

2.1 Multi-task FCN Architecture

The multi-task FCN accomplishes two tasks: computing a normalized centerline distance map and detecting branch endpoints. As shown in Fig. 2, the two tasks share the same encoder layers consisting of convolution (Conv), batch normalization (BN), ReLU activation function and max pooling (shown as ‘Down’ in Fig. 2) operations. The two tasks then have different decoder layers tailored for each task, consisting of conv, BN, ReLU, and upsampling operations. Skip connections are applied at the same scale between encoder and decoder layers to make effective use of both high-level and low-level features similar to [6]. Suppose a volumetric segmentation mask is $I^{X \times Y \times Z}$, and $\mathcal{I} = \{(x, y, z) | x \in \{1, \dots, X\}, y \in \{1, \dots, Y\}, z \in \{1, \dots, Z\}\}$ denotes the set of all voxel locations in the image.

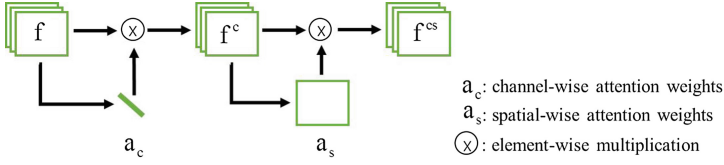


Fig. 3. Spatial-wise and channel-wise attention

Centerline Distance Map. A centerline distance map is defined as an image whose voxel intensity shows how close each voxel is to the nearest centerline point. Due to the large variations in branch radius (coronary artery proximal radius can be five times bigger than the distal radius), a straightforward Euclidean distance transform computation generates centerline distance map with largely variable range of values at different sections of the branch. To obtain a centerline consistently well-positioned in the “center” from beginning to end requires tricky balancing of cost image contrast between thick and thin sections.

To achieve the desired scale-invariance property, we propose to use FCN to generate a *locally normalized* centerline distance map. More specifically, during training, a local cross-sectional view of the segmentation mask perpendicular to the centerline tangent direction at each centerline point is obtained. Suppose the set of all foreground voxels in the cross-sectional view is \mathcal{S} . Then the locally normalized distance map value for voxel index $i = (x, y, z) \in \mathcal{S}$ is computed as

$$d_i = \frac{d_i^{Euc}}{\max_{i \in \mathcal{S}} d_i^{Euc}} \quad (1)$$

where d_i^{Euc} is the Euclidean distance of voxel i to the centerline point on the view. To further highlight contrast at portions closer to centerline, log-transform is applied to generate the reference centerline distance map

$$\mathbf{Y}_c = \log(\mathbf{D}_c + \delta) \quad (2)$$

where $\mathbf{D}_c = \{d_i | i \in \mathcal{I}\}$ is the locally normalized centerline distance map throughout the whole segmentation mask image I and δ is a small positive constant to avoid numerical issues. Note that the distance computation in Eqs. (1) and (2) is only carried out when generating training reference standards. During testing phase, the FCN will directly predict centerline distance map $\hat{\mathbf{Y}}_c$.

Spatial-Wise and Channel-Wise Attention for Centerline Distance Map. Traditionally the convolutional features at different spatial locations and channels are treated equally by the following layers. However, centerline extraction is inherently a localized task. Specifically, the narrow region surrounding the underlying centerline requires most attention for best discriminating contrast. Thus, a spatial-wise attention module is proposed to weight

feature maps at different spatial locations. Similarly, different channels of the feature maps can highlight different regions (some channels may focus more around centerline, while other channels may focus more on object mask boundary, etc). It also makes sense to add a channel-wise attention to weight different channels accordingly. Figure 3 shows the proposed spatial-wise and channel-wise attention module, inspired by [8]. The feature map \mathbf{f} is first weighted by channel-wise attention, generating \mathbf{f}^c , and then weighted by the spatial attention, generating \mathbf{f}^{cs} .

Channel-wise attention weights different channels by vector $\mathbf{a}_c \in \mathbb{R}^C$, where $\sum_{i=1}^C \mathbf{a}_c(i) = 1$. To obtain this weighting vector, average pooling is first applied on each channel of the feature map to obtain a summarized channel feature vector $\mathbf{v} \in \mathbb{R}^C$. Then a convolutional layer and a ReLU nonlinearity $\sigma(\cdot)$ are added to obtain the raw attention weights $\mathbf{u} \in \mathbb{R}^C$. In Eq. (3), $*$ is the convolution operator. \mathbf{W}_c is the convolutional kernel and \mathbf{b}_c is the bias vector. A softmax function is applied on the raw attention weights \mathbf{u} to obtain the final channel-wise attention vector \mathbf{a}_c , as shown in Eq. (4).

$$\mathbf{u} = \sigma(\mathbf{W}_c * \mathbf{v} + \mathbf{b}_c) \quad (3)$$

$$\mathbf{a}_c(i) = \frac{e^{\mathbf{u}(i)}}{\sum_{c=1}^C e^{\mathbf{u}(c)}}, \quad i \in \{1, \dots, C\} \quad (4)$$

To apply the channel-wise attention weights, the input feature map at channel i is multiplied by attention weight $\mathbf{a}_c(i)$ to obtain channel-weighted feature map

$$\mathbf{f}_i^c = \mathbf{a}_c(i) \cdot \mathbf{f}_i, \quad i \in \{1, \dots, C\} \quad (5)$$

Spatial-wise attention weight matrix $\mathbf{a}_s \in \mathbb{R}^I$ is obtained similar to the channel-wise attention. The raw spatial attention map \mathbf{q} is computed by applying a $1 \times 1 \times 1$ convolutional layer \mathbf{W}_s with bias \mathbf{b}_s and a ReLU nonlinearity $\sigma(\cdot)$ to \mathbf{f}^c (Eq. (6)). Then a softmax function is used to obtain the final spatial attention weights \mathbf{a}_s , where $\sum_{i \in \mathcal{I}} \mathbf{a}_s(i) = 1$. The spatial attention weight is applied by multiplying $\mathbf{a}_s(i)$ with features \mathbf{f}_i^c at location i (Eq. (8)).

$$\mathbf{q} = \sigma(\mathbf{W}_s * \mathbf{f}_c + \mathbf{b}_s) \quad (6)$$

$$\mathbf{a}_s(i) = \frac{e^{\mathbf{q}(i)}}{\sum_{j \in \mathcal{I}} e^{\mathbf{q}(j)}}, \quad i \in \mathcal{I} \quad (7)$$

$$\mathbf{f}_i^{cs} = \mathbf{a}_s(i) \cdot \mathbf{f}_i^c, \quad i \in \mathcal{I} \quad (8)$$

Branch Endpoint Detection. Different from centerline distance map which consists of continuous values inside the whole segmentation mask, branch endpoints are just a few isolated points. Directly predicting these points using a voxel-wise classification or segmentation framework is not feasible due to the extreme class imbalance. To tackle the class imbalance problem, a voxel-wise endpoint confidence map is generated by constructing a Gaussian distribution around each endpoint to occupy a certain area spatially. The FCN is then trained

to predict the endpoint confidence map, which has a more balanced ratio between nonzero and zero voxels.

Specifically, a spatial Gaussian field is generated around each endpoint

$$\mathbf{Y}_e(i) = \frac{1}{\sqrt{2\pi}\Delta} e^{-\frac{\mathbf{D}_e(i)^2}{\Delta^2}}, \quad i \in \mathcal{I} \quad (9)$$

where $\mathbf{D}_e(i)$ is the geodesic distance from voxel i to the nearest branch endpoint inside the segmentation mask. Δ controls the scale of the Gaussian field.

In testing phase, the predicted endpoint confidence map is thresholded by half of the maximum possible value, i.e., $0.5/\sqrt{2\pi}\Delta$. The centroid of each connected component is then returned as branch endpoints.

Loss Function. The loss function shown in Eq. (10) consists of two terms, one for the centerline distance map prediction and the other for the branch endpoint detection. We enforce the loss function to only account for regions inside the segmentation mask. Suppose the segmentation mask is $\mathbf{A} \in \{0, 1\}^{X \times Y \times Z}$. $\mathbf{A}(x, y, z)$ is 1 for every voxel (x, y, z) inside the segmentation mask, and 0 otherwise.

$$L = \gamma \|\mathbf{A} \odot (\mathbf{Y}_c - \hat{\mathbf{Y}}_c)\|^2 + (1 - \gamma) \|\mathbf{A} \odot (\mathbf{Y}_e - \hat{\mathbf{Y}}_e)\|^2 \quad (10)$$

In Eq. (10), \mathbf{Y}_c and $\hat{\mathbf{Y}}_c$ are the reference standard and the predicted centerline distance map. \mathbf{Y}_e and $\hat{\mathbf{Y}}_e$ are the reference standard and the predicted endpoint confidence map. \odot denotes the Hadamard matrix product operation, which is element-wise multiplication of two matrices. γ is a weighting factor that balances losses of centerline distance map and branch endpoint confidence map.

2.2 Minimal Path Extraction

Given a root point, a list of branch endpoints, and the underlying centerline distance map, a minimal path algorithm is used to extract the centerline of a tree-structured object.

We construct an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where set \mathcal{V} contains all vertices corresponding to voxels in the segmentation mask and set \mathcal{E} includes all edges connecting two neighboring vertices in set \mathcal{V} under a 26-neighborhood setting. On each vertex v_i , weight $w_{v_i} = \exp(\hat{\mathbf{Y}}_c(i))$ is set according to the centerline distance map. Note that each vertex carries a weight that is smaller when the corresponding voxel is closer to the centerline location, and larger when it is farther away from centerline. Given a starting vertex $s \in \mathcal{V}$ and an ending vertex $t \in \mathcal{V}$, a minimal path between the two is defined as $\mathbf{p} = (p_1, p_2, \dots, p_K)$, $p_k \in \mathcal{V}$, $k = \{1, 2, \dots, K\}$ such that (1). $p_1 = s$, $p_K = t$; (2). every two neighboring vertices in the path is connected by an edge; (3). the sum of vertex weights along this path is minimized (Eq. (11)).

$$\mathbf{p} = \underset{(p_k, p_{k+1}) \in \mathcal{E}}{\operatorname{argmin}} \sum_{k=1}^K w_{p_k} \quad (11)$$

Such a minimal path from s to t corresponds to the desired centerline between the two points. To extract the centerline of a tree-structured object such as a coronary artery tree, one root point usually correspond to multiple branch endpoints t_1, t_2, \dots, t_T . In this case, the minimal path between root point s and each endpoint $t_i, i \in \{1, 2, \dots, T\}$ is computed respectively. Then we trace each path from the end to the start point sequentially. Once the current path intersects with some previously traced paths, it is merged into the previously traced paths. The centerline points are finally smoothed by an iterative mean filtering in a small window for smoother appearance.

3 Experiments and Results

3.1 Experimental Design

To evaluate the proposed method, 620 volumetric coronary CTA scans of 620 patients are used. The image spacing is first normalized to $0.4 \times 0.4 \times 0.4 \text{ mm}^3$. Coronary arteries and ascending aorta are segmented by a semi-automatic software with manual review and editing. The segmentation masks of coronary arteries and ascending aorta serve as input to the experiment. Since coronary artery originates from ascending aorta, the root points of each coronary vessel tree are readily available as the artery voxels connected to aorta. To simplify notation, we use CL as a shorthand for ‘centerline’.

Manual annotations of centerline are hard to obtain due to the complex 3D structure of vessels and the single-pixel-wide requirement. Thus, during training, centerlines extracted by a state-of-the-art traditional method (called *baseline*) serve as the training reference truth for DeepCL. During testing, the degree of matching between DeepCL and baseline is first studied as a sanity check. Then various metrics requiring no “truth” centerline such as centerline to segmentation mask Hausdorff distance, and independent human expert review, are utilized to evaluate DeepCL and baseline method.

Baseline. The baseline method is also a minimal path approach. However, both branch endpoints and centerline distance map are computed by traditional methods. The centerline distance map \mathbf{D}^{Sig} is derived from the Euclidean distance map \mathbf{D}^{Euc} of the segmentation mask by summing three sigmoid functions to highlight the contrast in regions close to the centerline area.

$$\mathbf{D}^{\text{Sig}} = \sum_{i=1}^3 \frac{1}{1 + \exp(-\frac{\mathbf{D}^{\text{Euc}} - \beta_i}{\alpha_i})} \quad (12)$$

Three pairs of parameters (α_i, β_i) , each controlling the width and level of a contrast window, are tuned to enhance central contrast for vessel segments with large, medium, and small diameters respectively. Summing of these three sigmoid functions results in a relatively good contrast around centerline area throughout the whole vessel tree. The branch endpoints are detected as local maxima of

the arrival time by a breadth first search, starting from the root point at the junction of aorta and artery to each voxel throughout the artery segmentation mask. All related parameters are tuned on a different dataset.

DeepCenterline. We randomly divided 620 scans into three dataset: 200 scans for training, 20 scans for validation and 400 scans for testing. On the training set, centerlines extracted by the baseline method are used as the reference truth. Although generated by a strong baseline method, the reference truth still contains errors such as missing branches, wrong bifurcations in case of imperfect segmentation mask, etc. The locally normalized centerline distance map and the branch endpoint confidence map are generated based on the reference truth. The parameters of the proposed method are tuned based on the validation set. The tuned model is applied to the 400 testing scans to evaluate the performance.

The input patch size is $64 \times 64 \times 64$ voxels. The standard deviation of Gaussian field Δ in Eq. (9) is set to 3 mm. The loss weighting coefficient γ in Eq. (10) is 0.5. Our multi-task FCN network is optimized by stochastic gradient descent, with batch size of 3. Total number of epochs is 20. The initial learning rate is 10^{-2} , which is divided by a factor of 2 every 5 epochs.

Evaluation Metrics. Several evaluation metrics based on either objective metrics or independent human expert review are used for a thorough comparison of the performance of baseline method and DeepCL on the test set.

- i Mean centerline to centerline distance. The mean absolute distance from centerline A to centerline B is defined as the mean of the absolute distance to the nearest point on B for every point on A.
- ii Coverage percentage. A point on centerline A is covered by centerline B if the closest point on B is within half a voxel (0.2 mm).
- iii Number of missing endpoints. The number of endpoints not found by automated algorithm is counted manually in patient-level as well as branch-level.
- iv Number of scans with wrong bifurcations. This usually happens when two branches are spatially close or even briefly joined at a certain section. The centerline could wrongly consider this brief joining as a bifurcation.
- v Average patient-level centerline length. The patient-level centerline length is computed as the sum of lengths of each centerline segment. In general, the less straight “shortcut” centerline takes, and/or the more branch endpoints are detected, the longer centerline will be.
- vi Hausdorff distance. Hausdorff distance is defined as the maximum of distances from every artery segmentation mask voxel to the closest centerline point. Hausdorff distance shows how close each segmentation voxel is being covered by the extracted centerline.
- vii Overall success rate. A centerline extraction is called fully successful when an expert reviews the centerline and determines that the centerline covers all branches sufficiently, has no spurious false positive branch, no wrong bifurcation, and no obvious deviation from the center throughout all sections.

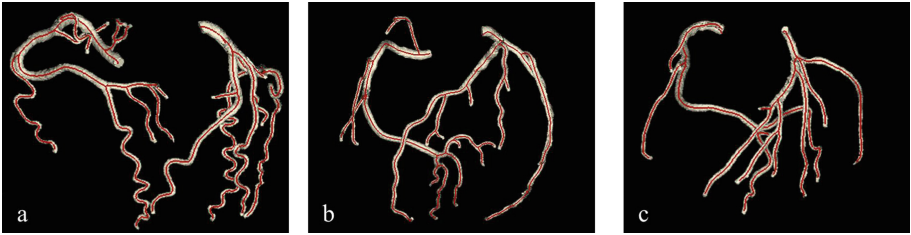


Fig. 4. Example coronary artery with: (a) very tortuous course; (b) small branches; (c) two spatially close branches. Red lines are centerlines generated by DeepCL. (Color figure online)

3.2 Results

Figure 4 displays three examples of coronary artery segmentation masks overlaid with centerlines extracted by DeepCL. For each coronary artery, radius changes substantially from the proximal to the distal side. Different coronary arteries have large variations of vessel curvature, shape and branch topology. Despite all these difficulties, our method is able to extract well-positioned centerline for all branches without false positive branches.

Table 1. Degree of matching between DeepCL and baseline method. A→B measures the distance from one point on centerline A to nearest point on centerline B.

	Baseline → DeepCL	DeepCL → baseline
Mean distance (mm)	0.066 ± 0.053	0.068 ± 0.014
Being covered by (%)	99.7 ± 0.7	99.5 ± 0.6

Matching to Baseline. Table 1 shows the degree of matching between DeepCL and baseline method. The mean centerline distance and “being covered by” percentage both shows how close/well one centerline is being covered by the other centerline. The low mean distance value and the very high coverage percentage on both direction (baseline to DeepCL and DeepCL to baseline) show that the two methods are in good alignment in general. However, a larger portion of baseline centerline points are being covered by DeepCL centerline (99.7%) by a smaller distance (0.066 mm) than the other way around (99.5%, 0.068 mm). This implies DeepCL provides slightly better coverage than baseline, which will be assessed in detail in the following analysis.

Notably, both DeepCL and baseline generate no spurious false positive branches in the extracted centerline according to visual inspection.

Performance Difference. Table 2 shows a detailed analysis regarding the difference between results generated by DeepCL and baseline. Bold items show the

Table 2. Difference in performance of DeepCL and baseline.

Metrics		Raw number #/#		Ratio %	
		Baseline	DeepCL	Baseline	DeepCL
Missing endpoint	Patient-level	170/400	34/400	42.5%	8.5%
	Branch-level	233/6048	35/6048	3.9%	0.6%
Scans with wrong bifurcation		28/400	11/400	7.0%	2.8%
CL length (mm)		308.9	314.3	-	-
Overall success rate		217/400	355/400	54.3%	88.8%

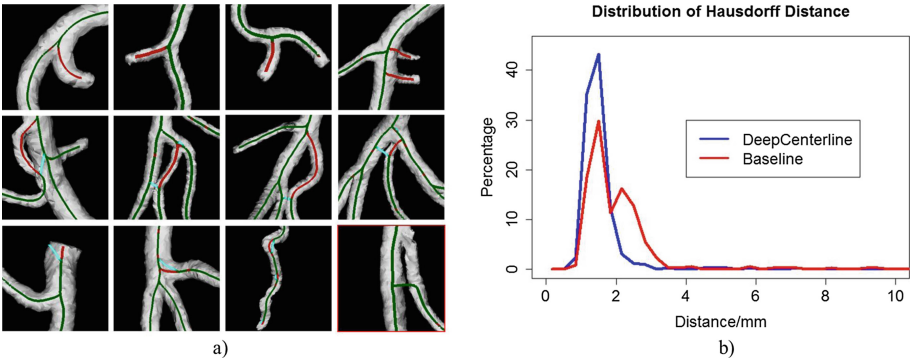


Fig. 5. Visual comparison of two methods and Hausdorff distance distributions. (a) Red is centerlines from DeepCL. Cyan is centerlines from baseline method. Green shows the overlap of both centerlines. First row: DeepCL finds branches missed by baseline method. Second row: DeepCL avoids wrong bifurcations generated by baseline. Third row: DeepCL generates centerlines well-positioned at central location, avoiding taking straight shortcuts at complex bifurcation regions or tortuous segments. The last figure with red border shows a failure case for both DeepCL and baseline. (b) Hausdorff distance distribution from voxels to the nearest centerline points for both methods. (Color figure online)

method with better performance on each metric. DeepCL finds more branch endpoints on both branch-level and patient-level. The number of wrong bifurcations shown in DeepCL is also less than that in baseline. Besides, DeepCL increased the average patient-level centerline length, due to finding of more endpoints and staying to the center instead of taking straight shortcuts at tortuous regions. Overall, the percentage of scans with successful centerline extraction without any type of error on any branch is substantially increased from 54.3% to 88.8%.

Figure 5(a) shows qualitative comparison of both methods. Compared to baseline, DeepCL shows significant improvement in finding more endpoints, reducing number of wrong centerline bifurcations at region with vessels close together, and staying at center instead of taking straight shortcut at regions with high curvature.

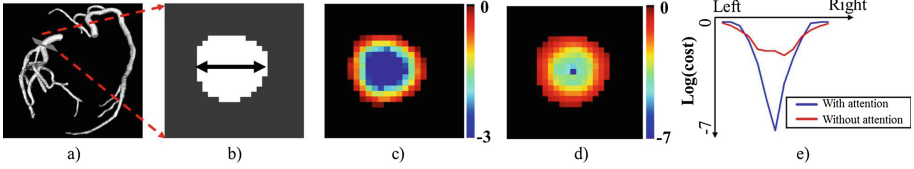


Fig. 6. Comparison of centerline distance map prediction with and without attention. (a) Coronary artery segmentation mask. (b) A cross-sectional view of segmentation mask. (c) Centerline distance map without attention module. (d) Centerline distance map with attention module. (e) Centerline distance map values at the profile line shown as double-arrowed line in (b). With attention, the centerline distance map shows a high peak around centerline instead of a plateau by the model without attention.

Figure 5(b) shows the distribution of patient-level Hausdorff distance from any segmentation mask voxel to centerline. A smaller Hausdorff distance means that *all* voxels in the segmentation mask are “covered” by a closer nearby centerline point. The majority of Hausdorff distances for DeepCL centerlines form a peak around 1.7 mm. In contrast, the baseline method has a longer tail towards higher Hausdorff distance values, with a significant percentage of scans having Hausdorff distance above 2 mm. This shows that DeepCL covers all segmentation mask voxels more closely.

Importance of Attention. Figure 6 compares the centerline distance maps generated with and without the attention module. As shown in Fig. 6(e), the centerline distance map has a clear peak around the central location by using attention. In contrast, the CL distance map generated by model without attention results in a “plateau” for a large area. If this situation occurred at regions with high curvature, then the minimal path extractor can easily pick a straight shortcut passing through non-center plateau points as the resulting centerline. This problem is greatly alleviated by utilizing the attention module to improve the centerline distance map contrast around real centerline point.

4 Discussion

The proposed method tackles multiple long-existing challenges of centerline extraction. A novel branch endpoint detection algorithm using Gaussian-field based endpoint confidence map is developed to detect the extremely sparse branch endpoints. The centerline distance map is made scale-invariant to the substantial diameter change of vessel branches from proximal to distal sections through local normalization within each cross-sectional view. The scale-invariant centerline distance map helps generate well-positioned centerline throughout all sections. Log-transform and attention module are utilized to further highlight the central locations, aiding accurate localization of the single-pixel-wide centerline. The large model capacity of FCN provides robustness to minor imperfections of segmentation masks.

Note that the reference “true” centerlines used in the training phase are results generated by the baseline method *without* manual correction, due to the difficulty of manual correction of a single-pixel-wide centerline. Despite of this disadvantage, our method achieves better performance than baseline on the test set. This shows the good generalization ability of the proposed model. A further study topic is to use our current method’s output as reference standard to train another FCN model. It would be interesting to see whether this second model would further improve upon the first-generation model.

5 Conclusion

We propose a novel centerline extraction framework which combines a multi-task FCN computing a locally normalized centerline distance map and detecting branch endpoints, with a minimal path extractor. The proposed method is the first deep-learning based centerline extraction method that guarantees single-pixel-wide centerline for a complex tree-structured object. Designed to be robust to substantial scale changes at different locations and minor imperfections of segmentation mask, the proposed method generates centerlines with more complete and closer coverage of segmentation masks without false positive branches.

Acknowledgement. The authors would like to thank Xiaoyang Xu and Bin Ouyang for organizing the dataset. This study has received funding by Shenzhen Municipal Government (KQTD2016112809330877).

References

1. Gülsün, M.A., Funka-Lea, G., Sharma, P., Rapaka, S., Zheng, Y.: Coronary centerline extraction via optimal flow paths and CNN path pruning. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9902, pp. 317–325. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46726-9_37
2. Jin, D., Iyer, K.S., Chen, C., Hoffman, E.A., Saha, P.K.: A robust and efficient curve skeletonization algorithm for tree-like objects using minimum cost paths. *Pattern Recogn. Lett.* **76**, 32–40 (2016)
3. Kirişli, H., et al.: Standardized evaluation framework for evaluating coronary artery stenosis detection, stenosis quantification and lumen segmentation algorithms in computed tomography angiography. *Med. Image Anal.* **17**(8), 859–876 (2013)
4. Metz, C., Schaap, M., Weustink, A., Mollet, N., van Walsum, T., Niessen, W.: Coronary centerline extraction from CT coronary angiography images using a minimum cost path approach. *Med. Phys.* **36**(12), 5568–5579 (2009)
5. Mirikharaji, Z., Zhao, M., Hamarneh, G.: Globally-optimal anatomical tree extraction from 3D medical images using pictorial structures and minimal paths. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10434, pp. 242–250. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66185-8_28

6. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
7. Xiong, G., et al.: Comprehensive modeling and visualization of cardiac anatomy and physiology from CT imaging and computer simulations. *IEEE Trans. Vis. Comput. Graph.* **23**(2), 1014–1028 (2017)
8. Zhang, X., Wang, T., Qi, J., Lu, H., Wang, G.: Progressive attention guided recurrent network for salient object detection. In: CVPR, pp. 714–722 (2018)
9. Zheng, Y., Tek, H., Funka-Lea, G.: Robust and accurate coronary artery centerline extraction in CTA by combining model-driven and data-driven approaches. In: Mori, K., Sakuma, I., Sato, Y., Barillot, C., Navab, N. (eds.) MICCAI 2013. LNCS, vol. 8151, pp. 74–81. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40760-4_10