Coronary Artery Centerline Refinement Using GCN Trained with Synthetic Data

Zhanqiang Guo[1,2], Yifan Zhang[1,2], Jianjiang Feng[1,2](✉), Eddy Yang[3], Lan Qin[3], and Jie Zhou[1,2]

[1] Department of Automation, Tsinghua University, Beijing, China
jfeng@tsinghua.edu.cn
[2] Beijing National Research Center for Information Science and Technology, Beijing, China
[3] UnionStrong (Beijing) Technology Co. Ltd., Beijing, China

Abstract. Coronary artery centerlines extraction from cardiac CT angiography (CCTA) is an important but challenging task. The popular U-net based coronary artery segmentation and thinning approaches rely on large number of labeled data and tend to produce noisy results. We proposed a graph convolutional network (GCN) for refining noisy centerlines outputted by U-net and developed a coronary artery tree synthesis approach for GCN pretraining. Experiments demonstrate that both modules led to improved performance.

Keywords: Coronary artery extraction · Graph convolutional network · Synthesis data · Coronary CT angiography

1 Introduction

Coronary artery disease (CAD) is one of the leading cause of death in the world. Coronary CT angiography (CCTA) is often applied to obtain coronary artery information because of its non-invasion and sensitivity [1]. In medical image analysis, extracting coronary artery centerlines from CT is the first step to evaluate the extent of plaque and stenosis area [2]. Many automatic (semi-automatic) centerlines extraction methods have been proposed.

Existing coronary artery centerlines extraction methods can be coarsely divided into three categories: shortest path, tracking based, and segmentation based. The first type of method computes the shortest path between the starting and ending points in vessel maps. The key to this method is to define an appropriate cost function, which is smaller for points on centerlines than those on other locations [1,3]. In recent years, algorithms based on deep learning are used to extract features of each point on centerlines [4]. However, these methods generally require numerous manual interactions, and there may be shortcuts off

This work was supported in part by the National Natural Science Foundation of China under Grants 61976121 and 82071921.

E. Puyol Antón et al. (Eds.): STACOM 2021, LNCS 13131, pp. 21–28, 2022.
https://doi.org/10.1007/978-3-030-93722-5_3

the true centerlines. The second type of method depends on iterative tracking, which means the location of the next point is determined according to the characteristics (the direction and radius of the blood vessel, etc.) of the points on the centerlines that have already been obtained [5,6]. These methods generally need additional processing for bifurcation points and they are sensitive to lesions in coronary arteries.

The third type of method segments the coronary artery first, and then extracts the centerline from the segmentation [7]. CNN, especially U-Net [8,9], is the mainstream method of blood vessel extraction in recent years, but its success depends on a large number of data with annotations. What's more, it is a time-consuming and expensive task to label vascular trees in CTs. A possible solution is applying synthetic data with annotations to training [7,10], but it is very challenging to generate realistic and diverse CCTA with groundtruth vessel trees.

In order to solve problems mentioned above, especially the massive demand for data, we propose a framework that can reduce the dependence on segmentation network, which mainly consists of two parts. We firstly use 3D U-net to obtain over-segmented results of coronary artery, and then refine the results to get centerlines via graph convolutional network (GCN). The task of coronary artery over-segmentation is relatively easy with a small amount of data. Compared with generating CCTA, it is more feasible to generate centerlines of coronary artery trees, and the synthetic centerline data can be used to improve the performance of GCN. An effective method of tree structure generation is to interpolate two existing tree structures via geometric and structural blending [11]. So we can complete the whole framework of extracting the coronary artery centerlines with a small amount of data.

The main contributions of this paper include: (1) we propose a GCN model to post-process the centerlines obtained by an over-segmentation network. (2) we propose an approach to generating pairs of training data by adding noise and fracture to synthetic centerlines, which can be regarded as labels. In this way, we augment the training data effectively which can be used to pretrain our GCN, so as to improve the network performance.

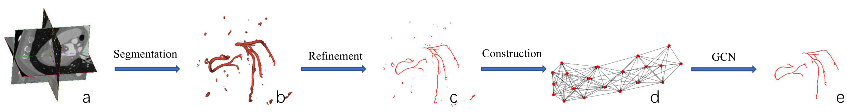

Fig. 1. Flowchart: CT image (a) is fed into the over-segmentation network to obtain segmented arteries (b); thinning image (c) is used to construct graph (d); (e) is the coronary artery centerlines after post-processing with GCN.

2 Method

The flowchart of the proposed algorithm is shown in the Fig. 1. The CT image is processed into the mask of the coronary artery by an over-segmentation network. Then, the result is refined to be a centerline image with noise and fracture, which is used to construct a graph. Finally, we can get the precise result through a GCN network. In this step, the synthetic data is applied to improve the performance of the GCN model. The detailed information about our framework is described as follows.

2.1 Over-Segmentation Network

Illustrated as Fig. 2, our segmentation network used in this framework is customized from the basic 3D U-Net network [8], which is generated with the help of nnUNet framework [12]. Similar to the standard version, our segmentation network consists of 3D convolution, max pooling, up sampling, and the shortcut connection from the down-sampling path layer to the up-sampling layer path. Considering that the resolution on each axis of the image differs, the step size $2 \times 2 \times 2$ is changed to $1 \times 2 \times 2$ in the first down-sampling, so that the resolution of each axis can be processed to approximately the same. In order to improve the performance of the network, loss is added for supervision in each step of up-sampling.

2.2 Graph Building

Consider an over-complete undirected graph $G_{in} = \{V, E\}$, which contains subgraphs of the coronary tree and some redundant nodes and edges, where V represents the set of nodes, and $|V| = N$ represents the number of nodes. For each node $i \in V$, its features are expressed as $x_i \in R^{F \times 1}$, and $X \in R^{F \times N}$ is the matrix of node features. E is the set of edges, and $(i, j) \in E$ indicates that there is an edge connection between nodes i and j. $y_{(i,j)} \in R^{H \times 1}$ stands for the features of edge (i, j). The task of extracting the coronary artery centerlines from

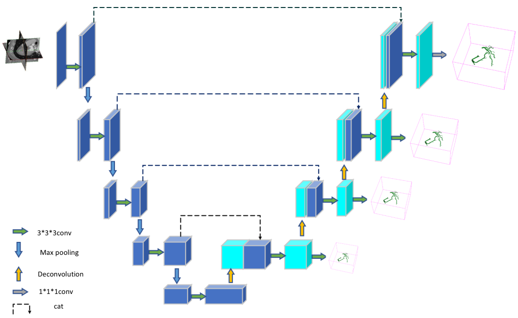

Fig. 2. 3D U-net based coronary artery over-segmentation

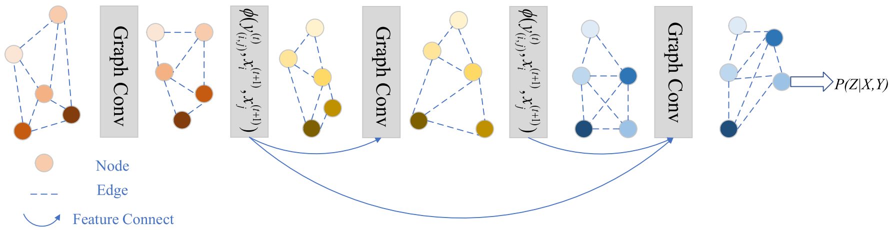

Fig. 3. GCN architecture: convolution of node features and full connection of edge features.

the image with noise and fracture is to train the model $f(\cdot)$, which can identify the points that belong to the centerline in the complete graph, $f : G_{in} \rightarrow G$, where G represents the graph of centerline.

The segmentation results in Sect. 3.1 can be refined to centerline images with noise and fracture, the points of which will be built to graphs. We select spatial features and pixel features for each point. The spatial features include the direction $[d_x, d_y, d_z]$, spatial position $[x, y, z]$ and two of the ten closest points with maximum and minimum distances $[d_{max}, d_{min}]$. The pixel features include the original image pixel value, the probability value of segmentation prediction and the value in the blood vessel enhancement image. For each point, it is connected with its nearest 10 points. The feature of each edge is defined as the reciprocal of the distance between involved two nodes.

2.3 GCN Architecture

The Graph Convolutional Network is shown in Fig. 3, which applies an alternate mode of updating node features and edge features. The updating process of node features is defined as:

$$x_v^{(t+1)} = \varphi(x_v^{(t)}, \{x_u^{(t)} : (u, v) \in E\}) \tag{1}$$

where $x_u^{(t)}$ is the features of all nodes connected to node v. $x_v^{(t+1)}$ and $x_v^{(t)}$ represent the features of node v at time $t+1$ and t respectively. The updating process of edge features is defined as:

$$y_{(i,j)}^{(t+1)} = \phi(y_{(i,j)}^{(t)}, x_i^{(t+1)}, x_j^{(t+1)}) \tag{2}$$

where $y_{(i,j)}^{(t)}$ and $y_{(i,j)}^{(t)}$ indicate the features of edge (i, j) at $t+1$ and t, respectively. $\phi(\cdot)$ is a fully connected network.

The output $P(z|i)$ of final network represents the probability that the node i belongs to the centerline. The loss is the cross-entropy loss, expressed as:

$$L = -\sum_{i=1}^{N} [z_i log(P(\hat{z}_i|X, Y, W)) + (1 - z_i) log(1 - P(\hat{z}_i|X, Y, W))] \tag{3}$$

where X, Y correspond to the features of nodes and edges. W represents the parameters of the network, and z represents the label of the node i.

2.4 Data Synthesis

We consider that centerlines and trees resemble the same structure. For a coronary artery tree, we classify its branches into three categories. We define the longest branch as the main tree, the branches connected to the main tree as primary branches, and the branches connected to primary branches as secondary branches (if existing). For two centerlines T and S in our dataset, we match their main trees, primary branches and secondary branches respectively. Empty branches will be added if one branch doesn't have pairing branch. In this way, we construct the pairing relationship between all the branches of T and S. For each pair of corresponding branches, we uniformly sample the same number of points and interpolate these points by B-spline respectively. After B-spline interpolation, we obtain a tuple B, containing the vector of B-spline coefficients. Thus, for a certain pair of branches t and s, two tuples B_t and B_s of the same size are obtained. Linear combination of B_t and B_s can generate intermediate state between t and s.

$$B_{new} = \alpha \cdot B_t + (1 - \alpha) \cdot B_s \quad (0 < \alpha < 1) \tag{4}$$

As α goes from 0 to 1, it illustrates a process that B_t converts to B_s. By adjusting the value of α, we get different B_{new}, which can be restored to different branches. Applying the above operation to each branch, finally we generate a complete coronary artery tree based on a certain α. The synthetic centerlines can be used as training data after adding noise and fracture, with the original centerlines as its pairing labels.

3 Experiments and Results

3.1 Evaluation Metrics

We evaluate the results based on the following indicators.

(1) The classification accuracy rate of GCN: the ratio of the number of nodes classified correctly to the number of total nodes, which shows the classification ability of GCN.

(2) The mean distance between centerlines:

$$d_{err} = \frac{\sum_{i=1}^{N_{seg}} min[d_E(c_i, C_{ref})]}{2N_{seg}} + \frac{\sum_{j=1}^{N_{ref}} min[d_E(c_j, C_{seg})}{2N_{ref}} = \frac{d_{FP} + d_{FN}}{2} \tag{5}$$

where C_{ref} is the reference centerline, C_{seg} is the prediction centerline. N_{seg} and N_{ref} represent the number of points on the prediction centerline and the reference centerline respectively.

(3) Coverage percentage: a point on centerline A is covered by centerline B, if the distance between this point and the closest point on B does not exceed the threshold. We calculate the coverage percentage of the labels and the predicted centerlines respectively.

Given that the predicted results of some branches may be longer than the marked results, which should not affect the evaluation in practice, these points do not participate in the calculation when computing the above indicators.

3.2 Results on Our Dataset

In the public CAT08 dataset [13], only four main centerline labels of the coronary arteries are given, which cannot reflect the performance of our method on small branches. So we conducted experiments on a dataset from a local hospital, which includes 47 CT volumes with segmentation labels. The label of centerline is obtained by thinning manually marked coronary artery label image. We randomly selected 32 data for training and 15 data for testing. Half of training data was used to train the segmentation network and all training data was used to train graph network.

Table 1. Performance comparison of four methods: GNN (the results on the airway dataset [14]), refining the segmented image (3D U-net), GCN with spatial features (S), GCN with spatial and pixel features (S+V). The accuracy of node classification (Acc), the distance of predicted centerline and label (d_{PF}, d_{FN}, d_{err}), and the coverage ($C_{predict}, C_{label}$) are reported.

	Acc(%)	d_{PF}(mm)	d_{FN}(mm)	d_{err}(mm)	$C_{predict}$(%)	C_{label}(%)
GNN	–	2.21 ± 0.46	2.87 ± 0.50	2.54 ± 0.58	–	–
3D U-net	–	3.21	1.57	2.39	89.37	72.74
GCN(S)	80.41 ± 0.01	1.57 ± 0.05	1.21 ± 0.04	1.39 ± 0.03	92.41 ± 0.05	77.61 ± 0.11
GCN(S+V)	80.34 ± 0.18	1.58 ± 0.07	1.29 ± 0.04	1.44 ± 0.05	90.58 ± 0.02	75.02 ± 0.12

As shown in Table 1, our approach has better performance in reducing fracture and noise on the centerlines. And the results show that the spatial feature plays a key role in the GCN model, because the pixel feature has been used in the previous segmentation network.

3.3 Results with Synthesis Data

We access the usefulness of pretraining with synthetic data by comparing various scores mentioned above. As shown in Table 2, when using pretraining, the performance on each indicator improves or remains basically unchanged. Since the synthetic data is mainly used for training GCN, the improvement of its classification accuracy can directly prove the effectiveness of synthetic data in this task. We achieve the accuracy rate of 80.41% for training from scratch without pretraining and 81.32% when pretraining on synthetic data.

Figure 4 shows that compared to the method of refining the segmentation result, our approach perfoms better in avoiding noise and fracture on the centerline. So it can deal with stenosis and plaques that may exist on CT images.

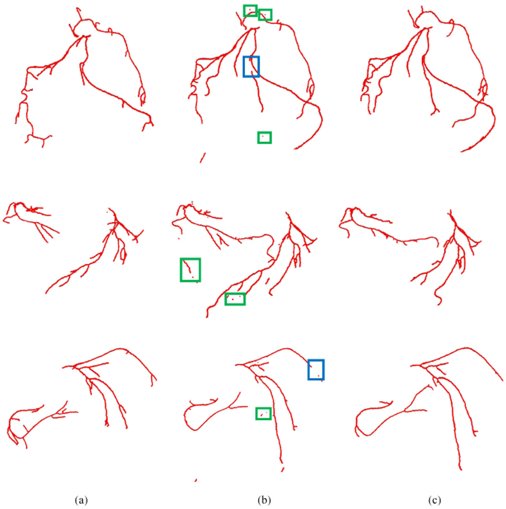

Fig. 4. Qualitative comparison of different algorithms on some examples. (a) annotated, (b) thinning image of over-segmentation result, (c) refining result of the proposed method. The green boxes and blue boxes respectively reflect the improvement of the noise suppression and fracture connection capabilities of our proposed method by using syntheic data. (Color figure online)

Table 2. Results of different data: the performance on synthesis data, the performance on real data and the results with pretraining.

	Acc(%)	d_{PF}(mm)	d_{FN}(mm)	d_{err}(mm)	$C_{predict}$(%)	C_{label}(%)
Synthesis data	96.10 ± 0.19	0.72 ± 0.07	0.17 ± 0.01	0.36 ± 0.04	96.30 ± 0.16	92.58 ± 0.33
Real data	80.41 ± 0.01	1.57 ± 0.05	1.21 ± 0.04	1.39 ± 0.03	92.41 ± 0.05	77.61 ± 0.11
Pretraining	81.32 ± 0.07	1.45 ± 0.04	1.19 ± 0.01	1.32 ± 0.2	92.83 ± 0.09	77.47 ± 0.10

4 Conclusion

In this paper, we propose a GCN-based centerline extraction framework. It consists of a segmentation network and post-processing steps. Since GCN performs better in the task of processing centerline images with noise, we just need an over-segmentation result, thereby reducing the dependence on segmentation performance. The experiments show that our method achieves good performance. In addition, synthetic data is used to further improve the effect of network. In our framework, the requirements for annotating data can be significantly reduced, which is also very helpful in practice.

References

1. Leipsic, J., et al.: SCCT guidelines for the interpretation and reporting of coronary CT angiography: a report of the society of cardiovascular computed tomography guidelines committee. J. Cardiovasc. Comput. Tomogr. **8**(5), 342–358 (2014)
2. Marquering, H.A., Dijkstra, J., de Koning, P.J.H., Stoel, B.C., Reiber, J.H.C.: Towards quantitative analysis of coronary CTA. Int. J. Card. Imaging **21**(1), 73–84 (2005)
3. Krissian, K., Bogunovic, H., Pozo, J., Villa-Uriol, M., Frangi, A.: Minimally interactive knowledge-based coronary tracking in CTA using a minimal cost path. Insight J. **1** (2008)
4. Guo, Z., et al.: DeepCenterline: a multi-task fully convolutional network for centerline extraction. In: Chung, A.C.S., Gee, J.C., Yushkevich, P.A., Bao, S. (eds.) IPMI 2019. LNCS, vol. 11492, pp. 441–453. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20351-1_34
5. Zhou, C., et al.: Automated coronary artery tree extraction in coronary CT angiography using a multiscale enhancement and dynamic balloon tracking (MSCAR-DBT) method. Comput. Med. Imaging Graph. **36**(1), 1–10 (2012)
6. Lesage, D., Angelini, E.D., Funka-Lea, G., Bloch, I.: Adaptive particle filtering for coronary artery segmentation from 3D CT angiograms. Comput. Vis. Image Underst. **151**, 29–46 (2016)
7. Tetteh, G., et al.: DeepVesselNet: vessel segmentation, centerline prediction, and bifurcation detection in 3-D angiographic volumes. Front. Neurosci. **14**, 1285 (2020)
8. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46723-8_49
9. Wang, Y., et al.: Deep distance transform for tubular structure segmentation in CT scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3833–3842 (2020)
10. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
11. Wang, G., Laga, H., Xie, N., Jia, J., Tabia, H.: The shape space of 3D botanical tree models. ACM Trans. Graph. (TOG) **37**(1), 1–18 (2018)
12. Isensee, F., Jäger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: Automated design of deep learning methods for biomedical image segmentation. arXiv preprint arXiv:1904.08128 (2019)
13. Schaap, M., et al.: Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. Med. Image Anal. **13**(5), 701–714 (2009)
14. Selvan, R., et al.: Graph refinement based airway extraction using mean-field networks and graph neural networks. Med. Image Anal. **64**, 101751 (2020)