

CAP 5516 Medical Image Computing Spring 2022 Project Report

Kyle Beggs

Department of Mechanical and Aerospace Engineering
University of Central Florida

kbeggs07@knights.ucf.edu

Abstract

1D models of the cardiovascular system are useful for surgical planning where time is of the essence due to their much reduced computational cost. These models are built from extracting the centerlines from 3D geometry. It is important for centerline extraction to not miss any branches as this can impact results and having a fully automated pipeline for simulation which includes centerline extraction is an attractive feature. State-of-the-art centerline extraction methods require user input and fail to capture branches often. This project seeks to further improve upon current approaches using Neural Networks for centerline extraction.

1. Problem Definition

In recent years there has been significant interest in developing a pipeline for cardiovascular surgical planning for patient-specific care via 3D Computational Fluid Dynamics (CFD) simulations [1]. The main barrier to the implementation of such a tool for surgeons to use is computational cost, despite sustained advancements in compute power. However, the success of many proposed surgeries such as coronary bypasses depends on macro-hemodynamic quantities that can be accurately simulated with 1D models effectively reducing the computational cost by a factor of hundreds compared to 3D. An equally important goal in building a pipeline for patient-specific cardiovascular surgical planning is to be fully automated. Recent research thrust in this direction aims for these goals but requires user input [2].

Centerline extraction can be performed via the grassfire or shrinking ball algorithms with no user input, but these often fail to achieve the robustness needed for the results to be directly used in simulation. They often have unphysiologically small vessel branches near bifurcations and struggle where there might be sudden widening or narrowing of vessels, which is common in diseased patients. Other approaches simulate a wave front by solving the Eikonal equation or compute a minimum cost path, however, these

require input of a seed and/or target outputs [3] [4].

In light of this, the goal of this project is to improve upon the current state of the art neural networks to segment the centerline of the vessel is desirable such that no user input is required and no further pre-processing is needed before performing 1D surgical planning simulations.

2. Related Work

The first attempt at extracting centerlines (at least claimed by the authors) was by Guo et al. who used a Fully Convolutional Network for extracting coronary artery centerlines [5]. Lenga et al. built a system to extract centerlines on rib bones using the feature map output from the network to perform an external extraction algorithm [6]. Although technically not using Deep Learning to output centerlines, it is included because it is fully-automated and that is a goal of this project. The next deep learning approach to extracting centerlines is DeepVesselNet where they attempt to segment the geometry, centerline, and bifurcation points [7]. They use a cross-hair filter for speeding up computation and introduce a new loss function based on cross-entropy but takes into account the high class imbalance as centerlines represent a very small fraction (< 3%) of the total pixels in an image. They also use angiogenesis models to generate synthetic data to train on since obtaining 3D vascular datasets along with centerline labels are nonexistent. The most recent attempt by Mostafa et al. first identifies a seed point and then narrows the input image to a CNN to the local area around that point. The output of the CNN is the next point on the centerline, which then becomes the new local area input into the CNN [8] - similar to the wave front technique pioneered by Antiga et al.

3. Methods

Due to the lack of cardiovascular datasets freely available, the only two datasets found are used. A synthetic dataset from the DeepVesselNet paper and a real dataset from the Vascular Model Repository [9]. The DeepVesselNet synthetic dataset has 136 images of size (325, 304, 600)

and will be useful in testing the robustness of the network as it has a large amount of bifurcations and variance in vessel size. A sample of the dataset is shown in .

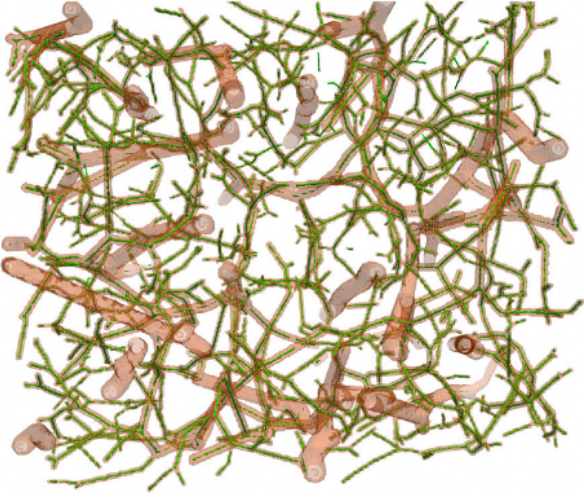


Figure 1. Example segmentation and centerline.

An architecture similar to that of Guo et al. [5] is used where segmentations (inputs) and distance maps (targets) are generated using the DeepVesselNet dataset. The distance maps are generated from the supplied centerlines as 1-pixel wide ‘lines’. The input to the network is the segmentation and output is a centerline distance map. This approach is used because complex and well performing segmentation networks exists, and so we want to begin from that point. The method should also be generalizable as its approach to finding branch endpoints does not rely on detecting particular anatomical features as in [8]. Endpoints are needed in the minimal path algorithm for final output of the centerline but represent a tiny fraction of the total voxels, suffering from major class imbalance. Guo et al. remedies this problem by generating a confidence map around the endpoint using a Gaussian distribution.

Two objectives will be investigated:

1. The use of strided convolutions vs pooling operations in the encoder part of the network.
2. What is the effect of training with perfect ground truth centerlines over truths generated with current state-of-the-art methods?

Objective two is a continuation of a question posed at the conclusion of Guo et al. because they generated the truth centerlines using current state-of-the-art methods (termed ‘baseline’) and then compared their network to those same baselines. Despite the conundrum of this approach, they still found their method was able to generalize better and increased accuracy of 30%! They propose using the better

output from their method to then train a second network and perhaps achieve even higher accuracies. Metrics that will be used to assess final performance: mean centerline to centerline distance, Hausdorff distance, and overall success rate which is defined 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.

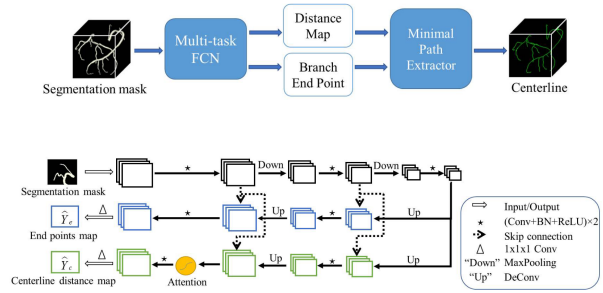


Figure 2. Network architecture used by Guo et al. [5] and the base for this work.

In this work, only the centerline maps are output and compared which should suffice for the objectives at hand. The centerline distance maps are generated via the binary distance transform which measures the distance from a pixel with 0 to a pixel with a 1. The $\log()$ transform is then computed of the map to further highlight the centerline. The final map along with the centerline is shown in 3.

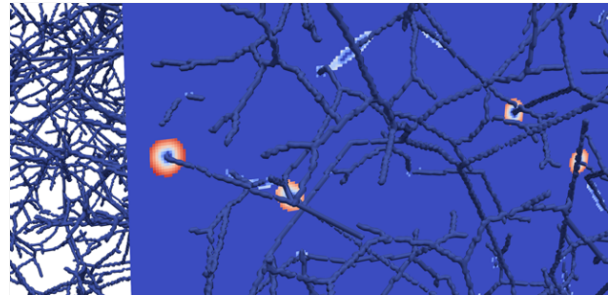


Figure 3. Centerline and centerline distance map.

The inputs images are randomly flipped among each axis and randomly cropped to a size of (300, 300, 300). The ADAM optimizer is used on a Mean Squared Error loss function with a Cosine Annealing Learning Rate Scheduler. The scheduler begins with a learning rate of $1e-3$ and the batch size is 6 and the model is trained for 100 epochs. It should be noted that the code for [5] was not provided and so the model was pieced together beginning from a Residual UNet based code.

Method	Train Loss	Val Loss	Time (hr)
Pooling	1.75e-3	1.75e-3	1.32
Striding	1.08e-3	1.07e-3	1.57
Pooling w/ Res.	1.41e-4	1.15e-4	1.33
Striding w/ Res.	5.54e-4	5.21e-4	1.31

Table 1. Results for your milestone and final reports.

4. Results

Unfortunately due to much time spent on initial debugging issues, only objective 1 was able to be investigated. Four models were compared - pooling, pooling with residual connections, striding, striding with residual connections. The training and validation loss data are presented in figure 4 and table 1.

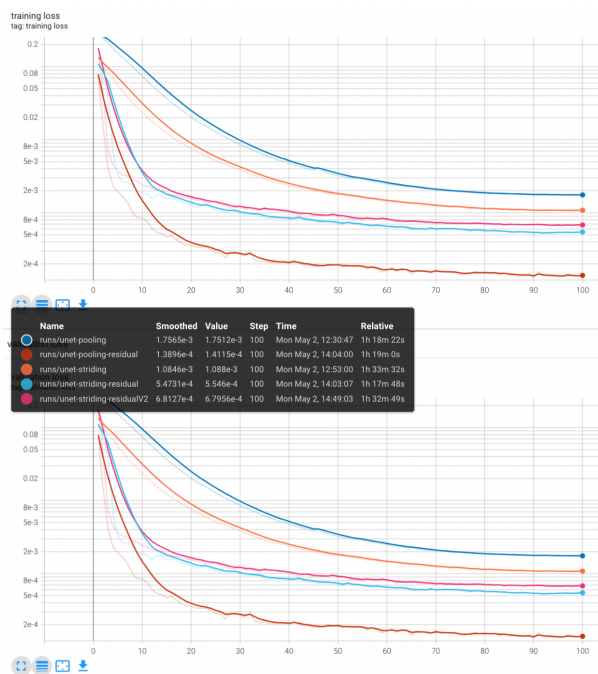


Figure 4. Training and validation loss curves.

Interestingly, when pooling operations are replaced with striding using a plain UNet (*no residual connections*) the striding improves performance, yet when residual connections are added, pooling is better. The target distance map is seen in figure 5 and the trained network outputs using pooling and striding can be seen in figures 6 and 7, respectively. It can be seen that the pooling does a slightly better job of localizing the centerline.

5. Conclusion

Unfortunately not much discussion is available due to limited results, but the pooling vs striding operations

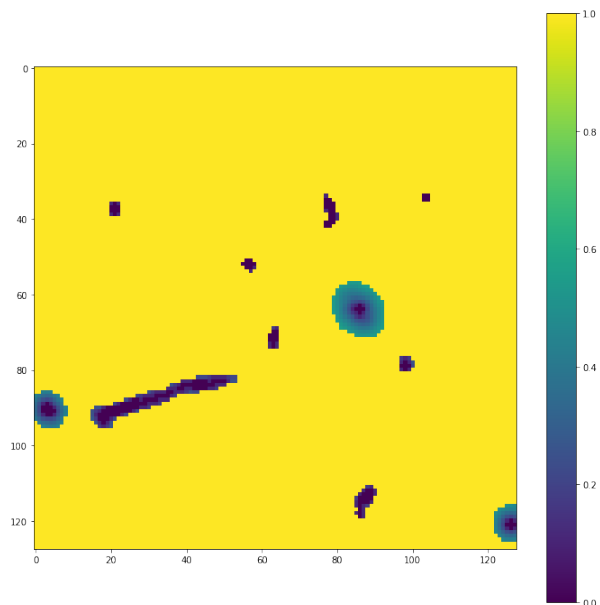


Figure 5. Target centerline distance map.

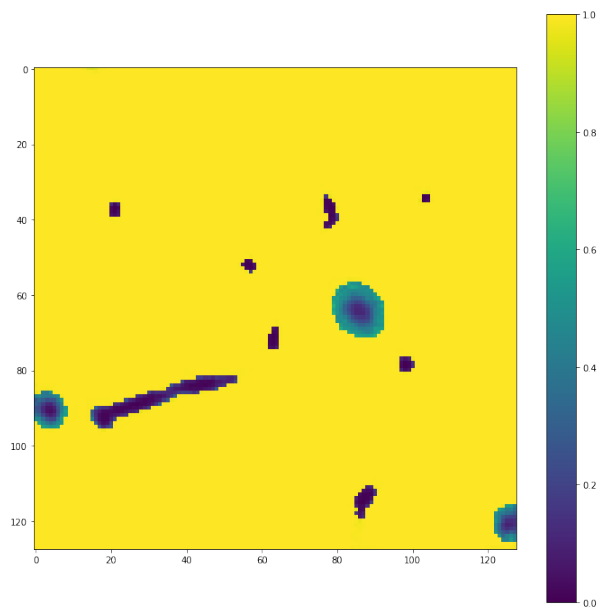


Figure 6. Network output using pooling with residual connections.

switching which performs better when residual connections are added is something interesting and should be investigated further.

References

- [1] T. M. Morrison, P. Pathmanathan, M. Adwan, and E. Margerrison, "Advancing Regulatory Science With Computational Modeling for Medical Devices at the FDA's Office of Science

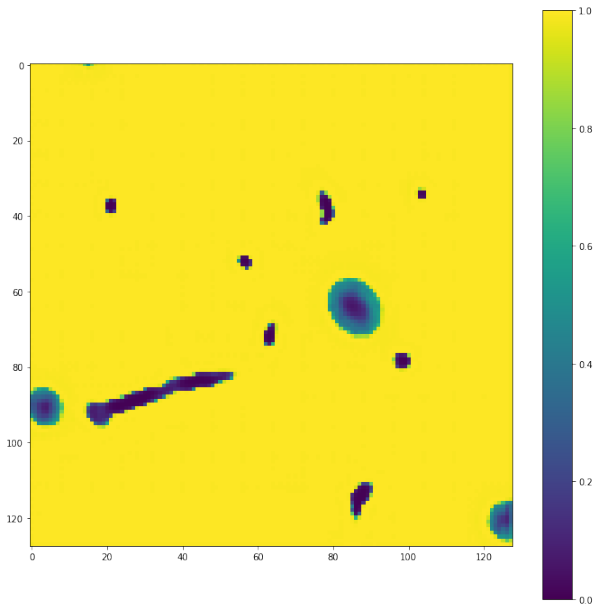


Figure 7. Network output using striding with residual connections.

nary Ostium Localization and Centerline Extraction Framework using Deep Learning,” in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, Nov. 2021, pp. 3846–3849. [1](#), [2](#)

- [9] N. M. Wilson, A. K. Ortiz, and A. B. Johnson, “The Vascular Model Repository: A Public Resource of Medical Imaging Data and Blood Flow Simulation Results,” *Journal of Medical Devices*, vol. 7, no. 4, p. 0409231, Dec. 2013. [1](#)
- and Engineering Laboratories,” *Frontiers in Medicine*, vol. 5, p. 241, Sep. 2018. [1](#)
- [2] M. R. Pfaller, J. Pham, A. Verma, N. M. Wilson, D. W. Parker, W. Yang, and A. L. Marsden, “Automated generation of 0D and 1D reduced-order models of patient-specific blood flow,” Nov. 2021. [1](#)
- [3] L. Antiga, B. Ene-Iordache, and A. Remuzzi, “Centerline Computation and Geometric Analysis of Branching Tubular Surfaces With Application to Blood Vessel Modeling,” Feb. 2003. [1](#)
- [4] D. Jin, K. S. Iyer, C. Chen, E. A. Hoffman, and P. K. Saha, “A Robust and Efficient Curve Skeletonization Algorithm for Tree-Like Objects Using Minimum Cost Paths,” *Pattern recognition letters*, vol. 76, pp. 32–40, Jun. 2016. [1](#)
- [5] Z. Guo, J. Bai, Y. Lu, X. Wang, K. Cao, Q. Song, M. Sonka, and Y. Yin, “DeepCenterline: A Multi-task Fully Convolutional Network for Centerline Extraction,” Mar. 2019. [1](#), [2](#)
- [6] M. Lenga, T. Klinder, C. Bürger, J. von Berg, A. Franz, and C. Lorenz, “Deep Learning Based Rib Centerline Extraction and Labeling,” in *Computational Methods and Clinical Applications in Musculoskeletal Imaging*, ser. Lecture Notes in Computer Science, T. Vrtovec, J. Yao, G. Zheng, and J. M. Pozo, Eds. Cham: Springer International Publishing, 2019, pp. 99–113. [1](#)
- [7] G. Tetteh, V. Efremov, N. D. Forkert, M. Schneider, J. Kirschke, B. Weber, C. Zimmer, M. Piraud, and B. H. Menze, “DeepVesselNet: Vessel Segmentation, Centerline Prediction, and Bifurcation Detection in 3-D Angiographic Volumes,” *Frontiers in Neuroscience*, vol. 14, p. 592352, 2020. [1](#)
- [8] A. Mostafa, A. M. Ghanem, M. El-Shatoury, and T. Basha, “Improved Centerline Extraction in Fully Automated Coro-