Predicting Blood Vessel Centerlines Using Convolutional Neural Networks

Numi Sveinsson
CS280 - Computer Vision
UC Berkeley
numi@berkeley.edu

Abstract

Blood flow simulations are proving to be an important part of cardiovascular research. A key bottleneck in modern simulation workflows is mesh generation. One solution is to automatically trace and segment the blood vessels using machine learning. The ability to predict the orientation of a blood vessel is crucial for automatic vasculature tracing. This paper explores the idea of using convolutional neural networks to directly predict a mapped centerline in the voxel grid. Results are promising but the accuracy is impacted by class imbalance in the training data. Different training data and loss functions are explored to solve this. Finally, potential next steps are discussed.

1. Introduction

Cardiovascular diseases are the leading cause of death in the United States. Computational blood flow simulations have become an important part of cardiovascular research. They play a crucial role in quantifying the flow, pressure and wall shear stress within healthy and damaged blood vessels, giving key insights for prevention and treatment planning of cardiovascular diseases.

In order to simulate flow through vasculature, researchers must first construct a surface mesh representing it. To do this, defining its domain in 3D space is necessary. This is most often done using medical image scans, e.g CT or MR images representing the patients body in 3D. Defining the blood vessel's domain requires segmenting these images for the vasculature of interest. One popular approach is to represent the blood vessels as 1D lines in 3D space connected at certain locations. These lines will be reffered to as centerlines for the purpose of this paper. Once the centerline is determined, the boundaries of the blood vessel can be determined in 2D perpendicular to it. These 2D contours can then be interpolated to form the 3D mesh to be simulated. The process of generating the blood vessel centerlines is called tracing (or tracking).

1.1. Blood Vessel Tracing

In the past, blood vessel tracing was mostly done manually preventing fast blood flow simulations. Recently, automatic blood vessel tracing has received increasing success [2]. These automatic tracing algorithms often depend on localized stepping, looking only at a sub-volume of the image data at each given time. Calculations are performed on these sub-volumes to determine the direction of the respective blood vessel in order to choose the next step to move to. In this work, we explore how convolutional neural networks may assist in determining the vessel orientation.

1.2. Convolutional Neural Networks

Convolutional Neural Networks, CNN for short, have proven immensely useful in many computer vision tasks. One of these tasks is medical image segmentation.

The U-Net architecture [3] from 2015 proved especially accurate for 2D medical image segmentation due to its U-shaped structure. Images have a very high dimensional feature space and the U-Net architecture downsamples it to lower dimensions using convolutions, capturing richer and richer spatial information in fewer and fewer dimensions. This part is called the encoder. This allows the network to encode information about the whole image in few dimensions, *e.g*. how different tissues lie with respect to each other. The second part of the U-Net is the decoder which upsamples this lower dimensional space back to its original resolution, ending with classification of pixels belonging to different anatomies. This means that only important information relating to segmentation is learned from the whole image and kept in the lower dimensional space which is then used for classification in the original resolution.

This paper aims at using this methodology to, instead of classifying individual pixels of the medical image, to learn a mapped blood vessel centerline directly.

1.3. Motivation

The aim of this work is to improve automatic blood vessel tracing based on localized stepping. That means to re-

turn useful information about the local sub-volume of interest at any given step of the algorithm. In essence, the best information to know is 1) the next point in 3D space to move to, to stay within the blood vessel, and 2) the size of the blood vessel in order to estimate the size of the next sub-volume as to encompass the blood vessel within.

This work hopes to solve the former problem. Instead of a single next point, the second best solution would be the centerline of the vessel in the sub-volume. From the centerline the next point can be estimated by taking either at the end of it or at some point along it.

Predicting the local centerline is not an easy task however. The idea of this paper is to train a CNN to learn a mapped version of the centerline in voxel space. This voxel based mapped centerline means that voxels belonging to the centerline have a high value and voxels further away from the centerline have lower values. The motivation is that by learning such a 3D voxel map of the centerline, that choosing the next point can be a task of going in the direction of lowest gradient, given that you are already on the centerline. Taking the next step is therefore like walking along a ridge of the mountain, staying within voxels of high value (belonging to centerline) and keeping away from voxels of lower values (further from centerline).

2. Method

The methodology of this work is divided into the following: 1) the dataset and sampling used to create sub-volumes for training, 2) how the centerline was mapped onto the voxel grid within each sub-volume, 3) the neural network architecture used, 4) the training procedure, and 5) some additional modifications made in order to improve the results.

2.1. Dataset and Sampling

The dataset used in this work belongs to another project, the Vascular Model Repository [5]. That includes 34 corresponding CT medical image volumes, 3D vasculature models and centerlines. An example of a vasculature case can be seen in Figure 1.

Since the tracing algorithm uses sub-volumes along the centerline, the training data for the neural network needs to represent that. The global 3D medical image volume was therefore sampled along the centerline of its corresponding vasculature model and saved. Two versions were extracted at each instance: 1) the original medical image data and 2) a binary segmentation of the sub-volume where 1s represent voxels within the blood vessel and 0s outside. The binary version is later used for the centerline mapping to create the ground truth used for training.

The samples varied in centering and size. Some were centered perfectly along the centerline while other were shifted perpendicular to it. The sub-volume sizes varied as well, from barely capturing the lumen of the vessel, to also including more 3D space around it. Both shift and size are sampled from Gaussian distributions. This variance was purposefully added to represent the variance that the tracing algorithm encounters in practice. When tracing a vessel, it does happen that the algorithm accidentally shifts away from the centerline, or over/under-predicts the vessel size. The neural network should be robust enough to that variance so that it still predicts an accurate centerline.

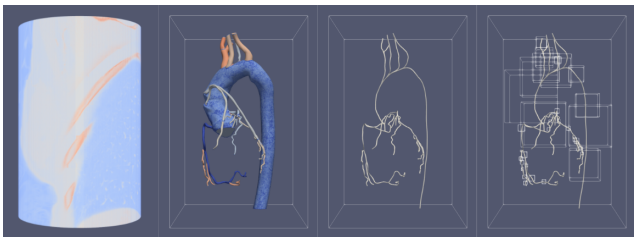


Figure 1. Dataset and Sampling. From left: medical image volume, 3D model of vasculature, centerlines of the 3D model, examples of sub-volumes sampled along the centerline.

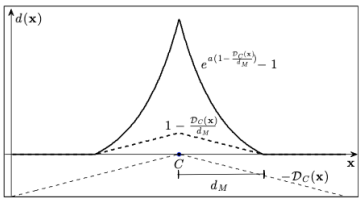


Figure 2. Centerline mapping. The equation used to create the ground truth for training. Image reused from [4].

2.2. Centerline Mapping

Representing a centerline in 3D voxel space can be done in many different ways. The most obvious one is binarization of all voxels that the centerline touches. However, Sironi *et al.* [4] found that encoding its information over more voxels proved easier for a neural network to learn. The idea being that a convolutional neural network is designed to represent well information that involves more spatial resolution, *e.g.* semantic segmentation that includes classifying each pixel. The centerlines of each sample was therefore mapped using Equation 1 where $D(x)$ is the distance a voxel has from the closest point on the centerline and d_M is the vessel radius.. Noteworthy is that the function value decreases exponentially with distance, down to 0 where the distance equals the radius of the vessel. Outside the vessel, voxels get the value 0. The value $a = 6$ was used.

$$G = \begin{cases} e^{a\left(1-\frac{D(x)}{d_M}\right)} - 1 & \text{if } D(x) < d_M \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

An example of a data sample showing the sub-volume

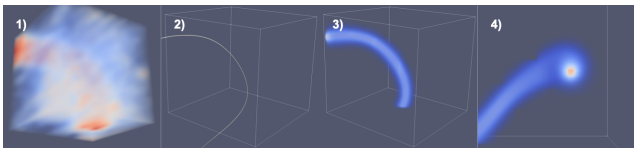


Figure 3. An example of a training data sample: 1) the image data, 2) the true centerline crossing the volume, 3-4) the resulting mapped centerline volume. The values range from blue (low) to red (high). All background has value 0 and is not shown.

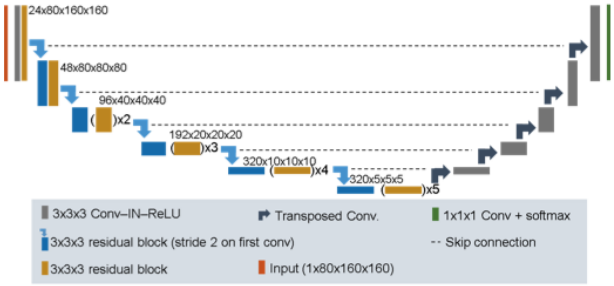


Figure 4. The 3D U-Net architecture used as introduced by Isensee *et al*. Image reused from the same paper [1].

image data, the centerline crossing through the volume and its mapping onto the voxel grid is shown in Figure 3.

2.3. Neural Network Architecture

The well-known U-Net was used for this work. A 3D version with additional modifications made by Isensee et al. was used [1]. The main difference being additional residual paths introduced within the architecture to counter-act vanishing gradients and allowing information to flow to each feature space more freely. The architecture can be seen in Figure 4.

2.4. Training

Training samples were in total 25k and 15% were used for validation. Testing samples were gotten from 5 vasculature models not used in training. Training took place using two GPUs on the Savio HPC cluster at UC Berkeley.

Two regression loss functions were initally used: mean squared error, MSE, and mean absolute error, MAE. The final decision of mean absolute value was made since it is prone to be more robust to outliers. In our case, some outliers outside the blood vessel may be ok since the main purpose is to achieve an accurate centerline representation of the blood vessel at large compared to the background around it.

2.5. Modification 1 - Normalization

The first modification to the initial work was to normalize the ground truth before training. If that it not done, then the ground truth proved to vary a lot between voxels and samples, based on how far the centers of the voxels were from the centerlines. Normalizing meant that all ground truth samples ranged from 0 to 1.

Version	Test Error
Original	2.8673
Normalized	0.0089
Normalized w/ Weighted Loss	0.0083

Table 1. Results on test set for the different versions. Error is mean absolute error.

2.6. Modification 2 - Weighted Loss

The second modification made was to weigh different voxels in the prediction differently when computing the loss. The reasoning being class imbalance. That means that in the ground truth, a very small proportion of the overall number of voxels belong to the vessel and have non-zero values. That means that when all voxels are treated equally for loss calculation, the background can overpower the loss compared to the non-zero values of voxels belonging to centerline. That can lead to the *default* value of predictions to be close to 0, since the neural network manages to get a low loss by predicting most of the voxels as background, since most of them were. This is not a new problem in medical image segmentation and is the motivation for other loss functions such as DICE loss which emphasizes the foreground more compared to background. DICE is however used in classification tasks and is therefore not applicable here.

To counteract this, a new loss function was proposed and tried. This loss function scaled the loss calculated at voxels within the blood vessel by double so that they would count more.

3. Results

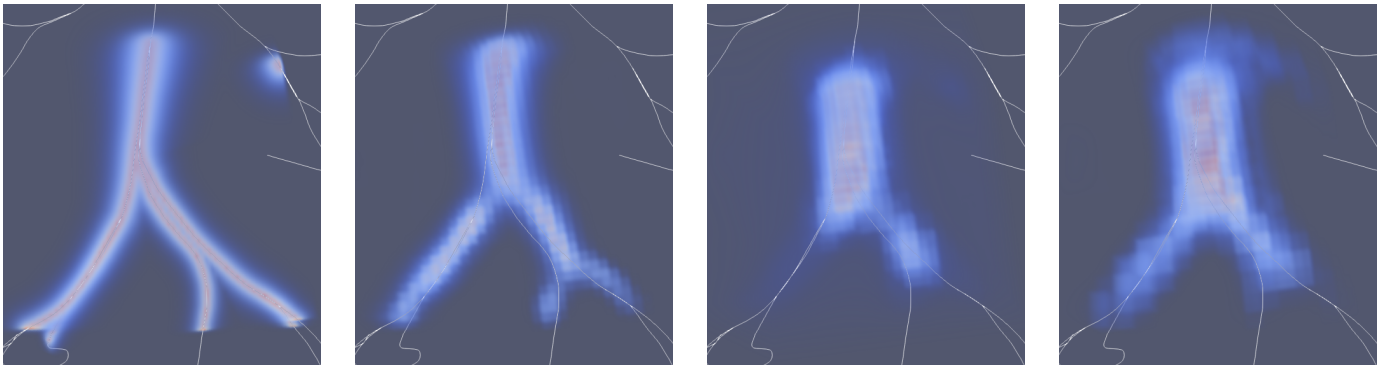
The results are shown for the three different versions:

1. The original implementation with mapping ranging from 0 to $e^6 - 1 = 402$ and loss is MAE.
2. Normalized version with the mapping ranging from 0 to 1 and loss MAE.
3. Normalized version with weighted loss function.

3.1. Quantitative

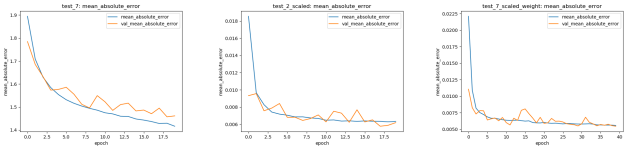
The test error results can be seen in Table 1. A graph of training loss for training and validation sets can be seen in Figure 6.

Figure 7 shows histograms of ground truth compared to predictions from the different versions.



(a) The ground truth. (b) The original. (c) Normalized data. (d) Weighted loss.

Figure 5. Prediction example, comparison for the different versions. The true centerline is shown as white lines.



(a) The original. (b) Normalized data. (c) Weighted loss.

Figure 6. Training loss plotted each epoch for the three versions. Validation loss is plotted in orange.

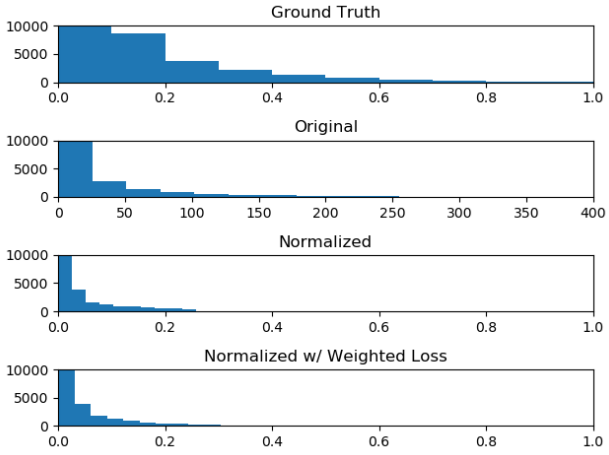


Figure 7. Histograms of values in the ground truth along with predictions from the different versions.

3.2. Qualitative

Some examples of predictions are shown in Figure 5. These results are important because of the eventual use of these prediction for automatic tracing.

Figures 8 and 9 show how well the first and third versions did predictions on a test model. This was done by random sampling along the centerline and predicting using those sub-volumes.

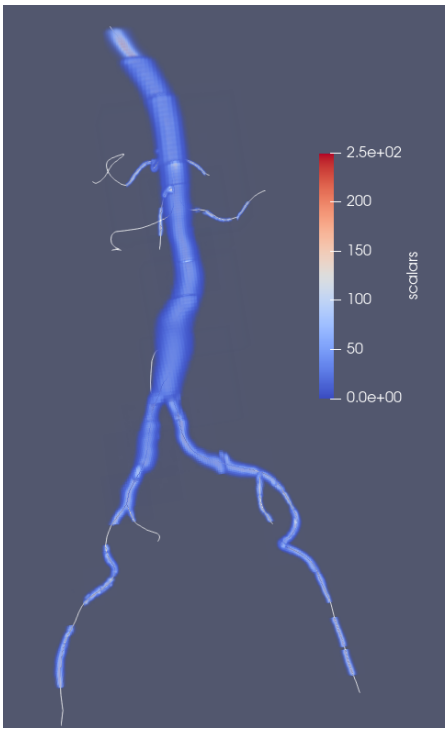


Figure 8. Resulting predictions from model version 1 on the image data sampled along the centerline of a test vasculature model. The true centerlines are shown in white.

4. Discussion

The results show promise of the 3D U-Net architecture to achieve low error values when learning to predict the centerline mapped volumes. This is shown in Table 1.

Another takeaway is the performance of the multiple versions on predictions. Despite low error rates for all versions, the original implementation seems to beat the other two qualitatively as seen in Figure 5. Figures 8 and 9 show the same thing. The original implementation seems to be

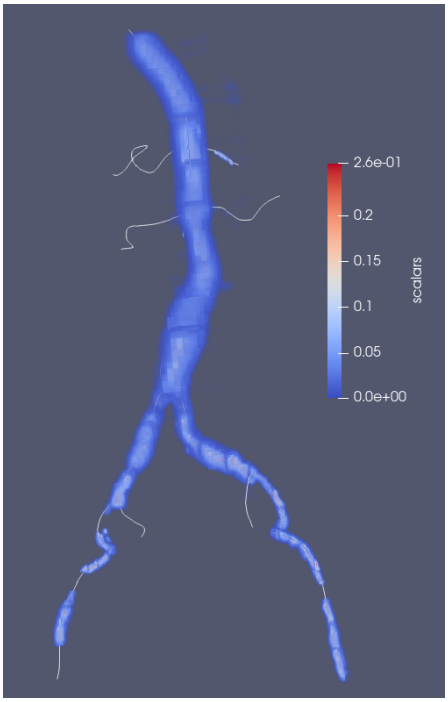


Figure 9. Resulting predictions from model version 3 on the image data sampled along the centerline of a test vasculature model. The true centerlines are shown in white.

more accurate and return cleaner predictions. Noteworthy is the difference in values for the two predictions. Figure 8 ranging to 250 but 9 to only 0.26. This difference might be caused by over-fitting in training for third version. This kind of difference in performance just because of normalization is surprising since the neural network should in theory be able to predict both functions.

Figure 5 shows the effect of increasing the weight on voxels within the vessel compared to background since that is the only difference between version 2 and 3.

Figure 7 shows the core of the problem at hand. The distribution targeted for learning by the U-Net is a severe long tail distribution. That makes sense because of how area increases with r^2 and that's how these values are determined. The key to good predictions will therefore entail a way to encourage the neural network to focus on the long tail in more detail. That can be done by further changing the loss function.

Potential next steps of this project are the following:

- Further changes to the loss function to emphasize the necessary distribution within the blood vessel over the background. Another idea is to incorporate a DICE loss based on all voxels over a certain threshold *e.g.* 100.

- Retrain using version 1 but with a weighted loss function.

- Exploring other neural network architectures, especially ones proven useful for regression tasks.

- Find a better metric for evaluation. For example to base it off of the gradients that the final prediction gives since that is what is going to matter in the end for tracing.

- Testing these results within the overall framework of vasculature tracing. Are these results good enough for localized stepping in the direction of lowest gradient?

- Explore different and better ways to represent bifurcating vessels. How can that information be encoded for a localized stepping tracing algorithm?

In conclusion, the 3D U-Net architecture proved capable, for the most part, to capture some of the centerline representation. Future work will need to focus on changing the loss function further to counter class imbalance as well as testing within a framework of localized stepping tracing algorithm to fully evaluate the effectiveness of this method.

References

[1] Fabian Isensee and Klaus H. Maier-Hein. An attempt at beating the 3d u-net, 2019. 3

[2] Dengqiang Jia and Xiahai Zhuang. Learning-based algorithms for vessel tracking: A review, 2020. 1

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 1

[4] Amos Sironi, Vincent Lepetit, and Pascal Fua. Multiscale centerline detection by learning a scale-space distance transform. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2697–2704, 2014. 2

[5] Nathan M. Wilson, Ana K. Ortiz, and Allison B. Johnson. The Vascular Model Repository: A Public Resource of Medical Imaging Data and Blood Flow Simulation Results. *Journal of Medical Devices*, 7(4), 12 2013. 040923. 2