End-to-End Image-to-Tree for Vasculature Modeling

Manish Sharma SKMANISH@GOOGLE.COM
Matthew C. H. Lee MATTHEWLEE13@IMPERIAL.AC.UK
James Batten J.BATTEN@IMPERIAL.AC.UK
Michiel Schaap MSCHAAP@HEARTFLOW.COM
Ben Glocker B.GLOCKER@IMPERIAL.AC.UK

Editors: Under Review for MIDL 2019

Abstract

Imaging can be used to capture detailed information about complex anatomical structures such as vessel trees. This can help to detect disease such as stenosis (blockages) which is important for diagnosis and clinical decision making. Current approaches for extracting vasculature from images involve generating binary segmentation maps followed by further processing. However, these binary maps may be sub-optimal, implicit representations of the underlying geometry while trees seem a more natural way of describing vasculature. In this work, we propose a novel *image-to-tree* approach, which is an end-to-end system for extracting explicit tree representations of vasculature from biomedical scans. We designed a moving patch algorithm that utilizes a U-Net component for predicting individual tree nodes. The methodology is presented for both synthetically generated tree images and publicly available Digital Retinal Vessel Extraction dataset (DRIVE). Using vascular tree construction, we discuss applications to thickness estimation in diabetic retinopathy prediction, and explore insights from visualizing these trees.

Keywords: Vasculature, tree extraction, retinal vessels, diabetic retinopathy, visualization

1. Introduction

The system of distributed blood vessels across each part of the body constitutes the vascular system, or vasculature. It is a crucial component to transfer oxygen and other nutrients to distant organs through blood circulation. The study of vasculature is important to characterize the flow of blood in lumens. Restricted blood flow through vessels can lead to severe health problems. Narrowing of vessels can be caused by lesions, calcification or plague reducing the area of the lumen. Aortic stenosis for example, restricts the blood flow from the left ventricle to aorta and increases the risk of heart failure (Czarny and Resar, 2014). Such information about blood vessels is captured in biomedical scans. Current approaches to extracting the vascular networks from these biomedical scans mostly involve segmentation followed by further processing (Bates et al., 2017), (Chapman et al., 2015). An elaborate survey of segmentation methodologies used was presented by Fraz et al. (Fraz et al., 2012). However, it seems sub-optimal to deal with binary segmentation maps while studying graph-like structures. We require a more interpretable and explicit way of modeling the underlying geometry.

In this work, we study an algorithm to directly retrieve a tree structure from an image, hence called *image-to-tree*. We propose a 'moving patch algorithm' that progressively moves over connected regions in the image, using a neural network at each region to construct nodes and edges of the tree. We assess the performance of several neural networks on a synthetically generated dataset and then apply it to biomedical scans. Synthetic data consists of vasculature-like images generated with a piecewise linear assumption and a kinematics modeling approach. We use the best performing models from tree construction in synthetic images to extract vascular trees from retinal images using DRIVE (Digital Retinal Vessel Extraction) dataset. Edges of vascular trees thus derived contain vital information about diabetic retinopathy in the arteriolar-to-venular width ratio (AVR) (Nguyen and Wong, 2009). In this direction, we also discuss an algorithm that estimates the width of each associated edge in the extracted vascular tree. We also present some insightful ways of visualizing the trees using GraphViz and node2vec.

A work that closely resembles our moving-patch approach for image-to-tree is the 3D vessel crawler by McIntosh and Hamarneh (2006). They propose a deformable object approach to crawl through 3D vessels. However, it is difficult to determine the prior shape knowledge. On the other hand, our approach is entirely data-driven, and hence easily extensible beyond vasculature construction. Discussion within this paper is organized as follows. In Sections 2 and 3, we explain and evaluate image-to-tree for synthetic and real data respectively. In Sec. 3.2, we discuss a method to predict edge thickness, corresponding to vessel widths. In Sec. 4, we show two visualization approaches for vascular trees.

2. Image-to-tree for synthetic images

Working with synthetic images to demonstrate image-to-tree is useful because of two reasons. First, its simplicity will help us to better understand the moving patch algorithm, and second, because we can generate ample data, we can better compare different approaches. We would like to generate images that have characteristics similar to that of a vascular system. In order to achieve this, we discuss two methods to generate synthetic images. We construct synthetic data, 2D images of size 51×51 pixels and consider trees which have edges of width one, further we root the tree in the center top of the 2D image.

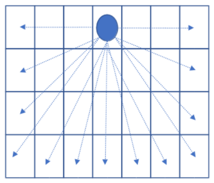
1. **Piecewise linear model** - In this approach, trees are grown downwards by successively sampling next node(s), starting from the root node based on a fixed spatial probability matrix $\mathbf{M}$ (Figure 1(a)). Value M_{ij} in this matrix denotes the prior of moving in the corresponding direction, assuming that current node is located at the top-center position. Moreover, at each node, there is an associated probability for conditions of split and terminate. It is also checked that the new edge does not create a cycle. Using Bresenham's line joining algorithm (Bresenham, 1965), we sketch the edges on a discrete pixel space, and add salt and pepper noise. A corresponding image generated using a depth first procedure is shown in Figure 1(b)

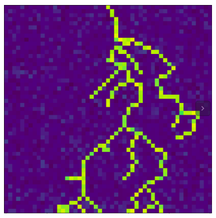
2. **Kinematics based model** - With the piecewise linear trees, it is difficult to model smooth vasculature-like bends. Here, we try to achieve this by assuming that the tree nodes are generated by a point vector with velocity and acceleration attributes i.e. $P_t = [p_x, p_y, v_x, v_y, a_x, a_y]$ (Fig 1(c)). The six values denote position, velocity and

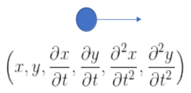
acceleration along the x and y axes respectively. At each point of edge generation, a condition is sampled from $\{\texttt{continue}, \texttt{split}, \texttt{terminate}\}$:

- `continue` - Point P_t moves to P_{t+1} with updated values of p_x, p_y, v_x, v_y
- `split` - Point P_t splits into P'_{t+1} and P''_{t+1} with values updated according to conservation of momentum and newton's third law of equal and opposite reaction.

Sample image using depth first procedure of edge generation is shown in Fig 1(d)


(a) Matrix M for linear model


(b) Sample image from model 1


$\left(x, y, \frac{\partial x}{\partial t}, \frac{\partial y}{\partial t}, \frac{\partial^2 x}{\partial t^2}, \frac{\partial^2 y}{\partial t^2}\right)$
(c) Kinematics model

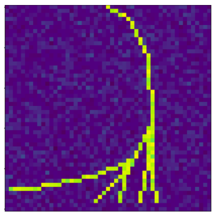

(d) Sample image from model 2

Figure 1: Two approaches for tree generation - piecewise linear model and kinematics model

2.1. Moving patch algorithm

Now that we have images which contain trees, we can define our algorithm to extract tree-structures from images. Figure 2 explains the moving patch algorithm. For synthetic images, the algorithm starts at the top center pixel of the image and samples a 11×11 patch. Then, it uses a predictor to output a 11×11 binary matrix, with ones corresponding to position of next node. This output is then used to sample the next image patch on which the predictor is run again and new node(s) are generated. The process continues until a matrix of 0s is reached, representing leaf state. In case of multiple predicted nodes (step 3,4 in Fig 2), they are processed on a depth first basis, storing the patch states on a stack.

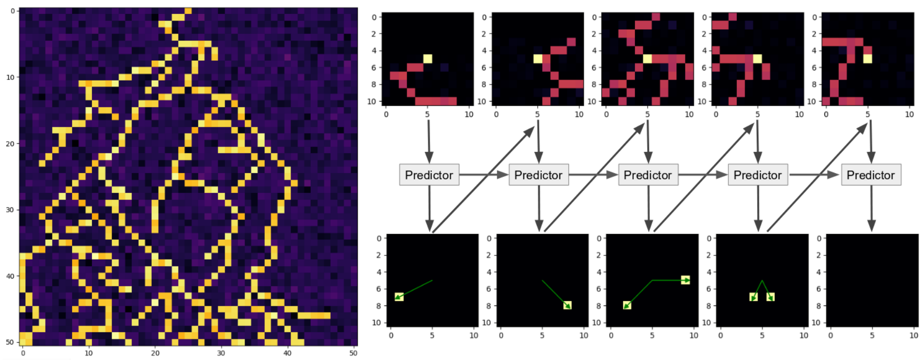
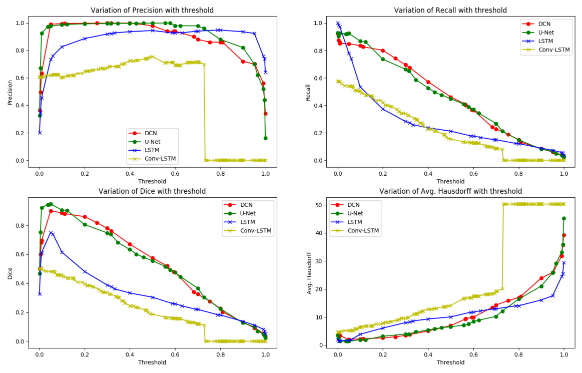


Figure 2: Explaining moving patch algorithm for a recurrent type predictor

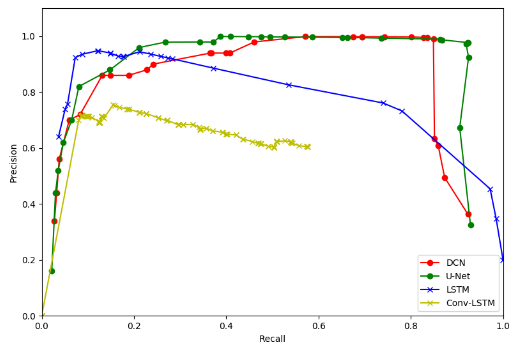
The predictor mentioned above is a deep neural network trained such that it predicts the position of the next nodes. Ground truth for such a predictor can be easily generated by using a depth first search on our original noiseless image. As a choice of predictor, we experiment with four types of neural networks - Long Short Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), Convolutional LSTMs (Xingjian et al., 2015), Deep Convolutional Network (DCN) and U-Net (Ronneberger et al., 2015). Former two networks use patch sequences as input (recurrent type), while the later two networks do not use sequence history.

2.2. Evaluation

In order to quantify the efficacy of the predictors, or how well the extracted tree represents the original tree there are several metrics which can be used to compare trees, here we are interested in a metric which is agnostic of number of nodes, edges and emphasizes spatial coverage. Since there is no single metric that captures all, we compute four measures - precision, recall, Dice score and average Hausdorff distance. We use these four metrics to evaluate the binary maps of tree generated by all four predictors.



(a) Performance metrics vs threshold



(b) Precision recall curve

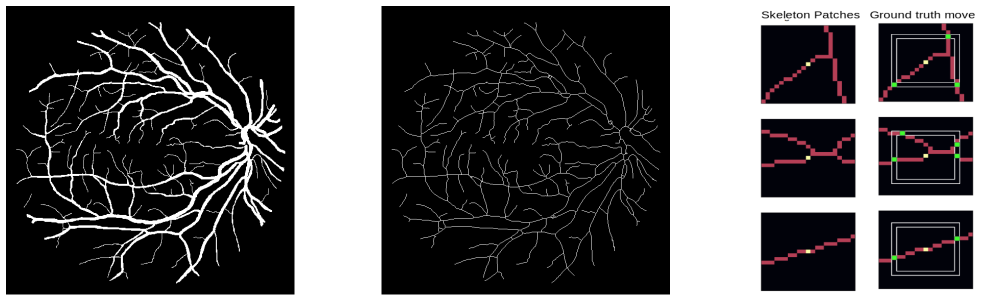
Figure 3: Sequence-based (LSTM, ConvLSTM) vs sequence-less predictors (DCN, U-Net)

The predictors output a real number in the range [0,1] for each cell in the image patch. A cutoff threshold is used to determine if a cell is predicted as a next node or not. Varying this threshold gives a tradeoff between high precision and high recall. Variation of precision, recall, Dice and haussdorff distance with threshold for all the models mentioned above is shown in Fig. 3(a). Precision recall curves (PRC) for four predictors are shown in Fig. 3(b). It can be learnt from both these figures that sequence-less models perform better on average. Specifically, U-Net shows the best performance from PRC analysis. Hence, we will use U-Net for tree extraction in the next section.

3. Image-to-tree for Digital Retinal Images Vessel Extraction (DRIVE)

Study of vasculature can help us determine pathology in blood vessels and discover potential stenosis/blockages. In our work, we focus on the retinal vasculature for mainly two reasons. First, blood vessels present on the thin lining at the end of the eye possess useful indicators for certain diseases like diabetes, macular degeneration, glaucoma and others, which could be useful for diagnosis by an optometrist. Second, the dataset from retinopathy is widely accessible for research allowing reproducibility of our results. In our experiments, we use the Digital Retinal Images for Vessel Extraction (DRIVE) dataset released by Staal et al. (2004) for benchmarking vessel segmentation algorithms. The DRIVE dataset consists of 40 retinal images. Each image also has a labeling for the vessels, generated under supervision of experienced ophthalmologists. Images are divided 20/20 into a training and testing set.

Ground truth for moving patch algorithm - For DRIVE dataset, we do not have the ground truth information for next move. While in section 2.1, it was easy to locate the point(s) of next move because trees were generated programmatically, this information is absent in the real data. Also, vascular trees do not have a uniform edge width everywhere. To overcome these issues, we use the available segmentation maps and define heuristics to get ground truth move(s) at each position. Two heuristics are sequentially used for this. Firstly, we skeletonize the segmentation. Since image-to-tree attempts to find the latent centerline tree, it is important to reduce the variable thickness vasculature into one-pixel wide skeleton. Morphological thinning (Lam et al., 1992) can be used to achieve this. Figure 4(b) shows the skeletonized version of the segmentation map in 4(a). After we have the one-pixel wide skeleton, we use an edge-heuristic to locate the point(s) of next move. Among several possible choices for this, our edge heuristic chooses the skeleton pixels crossing a window at Chebyshev distance of 5 pixels from the center point. This heuristic gave best results among others tried. Figure 4(c) shows the next points (in green) for 21×21 patches.



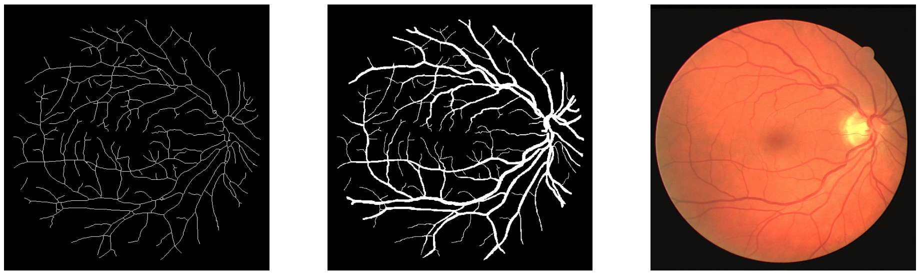
(a) Segmentation map (b) Skeletonized version (c) Edge heuristic

Figure 4: Process of ground truth extraction: Skeletonization followed by edge heuristic

3.1. Evaluation for three levels of vascular tree construction

We want to train our model to learn where to move next for a given input patch. Before training the models, it might be helpful to break our problem down into simpler levels, so that we can compare problem complexity and results from experiments. Therefore, we divide the more difficult problem into three levels of increasing order of difficulty. First, skeleton-to-tree is the task of extracting a tree data structure from the skeleton map (Fig. 5(a)). Second, segmentation-to-tree is the task of extracting a tree data structure using the segmented vessels (Fig. 5(b)). Similarly, retinal images to tree works on the raw retinal images as shown in Fig. 5(c). While the data of later two levels is directly available in DRIVE dataset, skeleton images are derived using morphological operations.

Model used for predicting the next node(s) in tree, or point(s) (corresponding to green pixels in 4(c)) for a patch of size 21×21 is a U-Net (Appendix A.1). Unlike single channel images of skeleton and segmentation, we add some preprocessing for the RGB retinal images.



(a) Skeleton map (b) Segmentation map (c) Retinal image

Figure 5: Three levels of image-to-tree problems in increasing order of difficulty

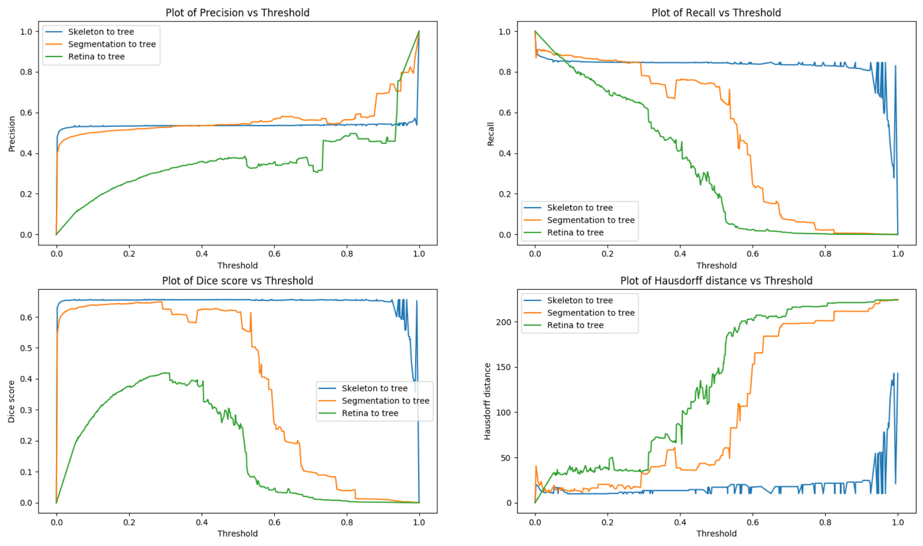
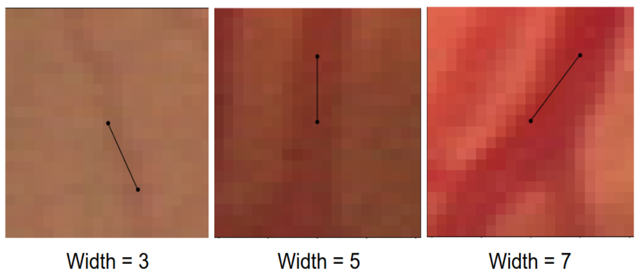
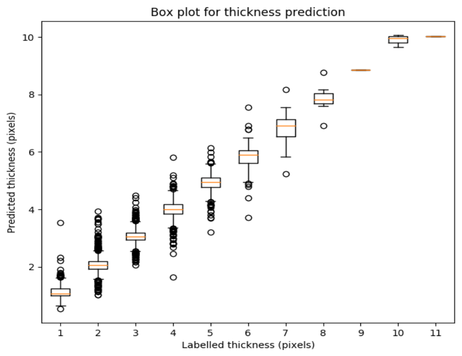


Figure 6: Performance metrics evaluated for all three levels of image-to-tree

Because of low signal to noise ratio in $21 \times 21 \times 3$ retinal patches, we perform image enhancement using histogram-equalization and histogram-specification (Osareh et al., 2002) techniques. It was experimentally observed that using 9 channels (3 RGB + 3 histogram equalized + 3 histogram normalized) improves overall system performance. 5K patches are randomly sampled from each of the 20 training images to obtain 100K patches overall. Data is augmented by rotations. Because we do not have the notion of a "right" tree, we use weak objective measures to compare different models and qualitative assessment for overall construction. The four measures we use are again - precision, recall, Dice score and Hausdorff distance. Fig 3.1 shows the comparison of these measures for the three levels. It can be seen that trees from retinal images (green) are objectively not as good as the trees from segmentation (orange) and skeleton (blue). Nevertheless, visually the extracted trees look very reasonable (Appendix A.2).



(a) Sample patches and width of vessels



(b) Box plot for thickness prediction

Figure 7: Predicting thickness of vessels

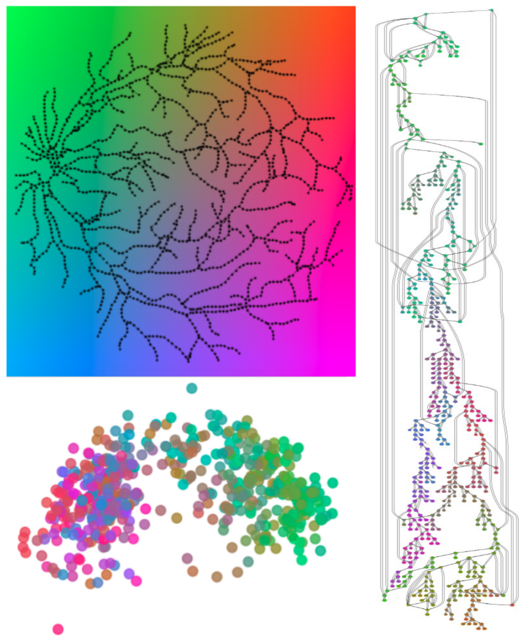
3.2. Predicting vessel thickness

Vessel width is an important marker for diagnosing Diabetic Retinopathy (DR). DR is among the major causes of loss of vision in people of working age. An important indicator is a low ratio of thickness of arteries to veins, also called AVR (Nguyen and Wong, 2009). Kondermann et al. (Kondermann et al., 2007) proposed a method to classify a segmented pixel as belonging to an artery or vein with an accuracy of 95.32%. In addition, we need to compute thickness of categorized vessels. We can use the segmentation annotations in DRIVE dataset to determine a ground truth thickness measure, against which we can train a regressor. The number of thinning operations which reduces an edge in a segmentation patch to an edge in skeleton patch is used as our thickness ground truth measure. Figure 7(a) shows some retinal image patches, along with the edges and determined thickness.

We train another U-Net that uses the same input patch as used for predicting the next edge. Additionally, another channel of predicted edge direction is used along with 9 other channels as before to train for corresponding edge's width. Figure 7(b) shows the boxplot of predictions. The root mean square error (RMSE) was roughly 0.1 pixel, implying a very good performance. This technique, used in conjunction with an arteries-veins classifier by Kondermann et al. (2007) could provide a good estimate of arteriolar-to-venular width ratio (AVR) and help detecting eye blindness in early stages and assist clinical prognosis.

4. Visualizing vascular trees and insights

Given that we can construct vascular trees with good accuracy, here we explore techniques to for visual assessment. The first visualization is based on graph embeddings using node2vec (Grover and Leskovec, 2016) which is a popular technique for generating node embeddings. The basic idea is that the lower dimension representations of a node must be similar to that of the nodes in its topological neighborhood. A biased random walk is used to explore the diverse neighborhood of a node for determining embeddings. Figures 8(a),8(b) (left-lower) show node2vec visualizations of position colored nodes in the extracted tree for two DRIVE images. This visualization can separate semi-vasculatures. We can also locate the point of central retinal artery (or optic nerve) node from the overlapping region between two halves.


(*a*) DRIVE test image 1

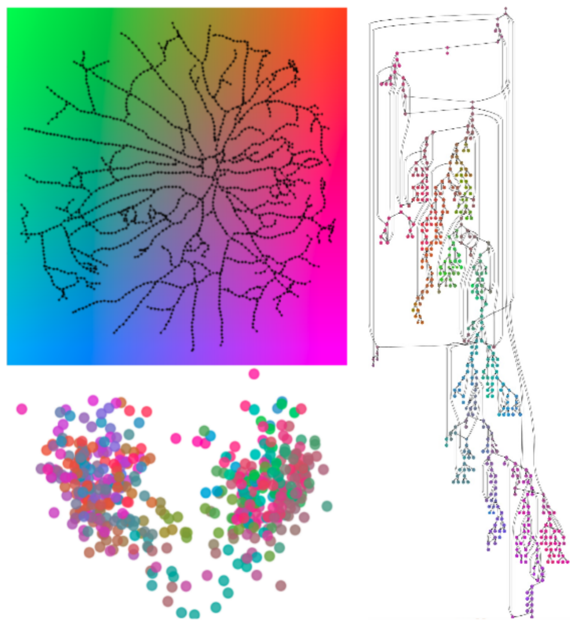

(*b*) DRIVE test image 4

Figure 8: Color coded nodes in original retinal image representation (left-top), GraphViz output (right) and node2vec output (left-bottom) for two images from DRIVE

The second visualization we explore is based on edge connectivity. This paradigm helps us visualize the branching structure by arranging the tree in a top down fashion.We perform two preprocessing steps. First, we cluster nearby detected nodes into one node (mostly in thick vessels). Second, we would like to avoid exceptionally long branches during visualization. Hence, we condense n-length edges into unit length. We use an open source tool called GraphViz (Ellson et al., 2001) to construct the tree creating a simplified layout for a graph structure by solving a linear integer program that optimizes for visual assessment. Fig 8 shows a flattened hierarchical arrangement produced by GraphViz. This can be used to visually separate subtree segments as arteries or veins or to visualize the flow of our moving patch algorithm, and hence model the directional flow of blood in lumens.

5. Conclusions

We propose a novel methodology to extract tree structures from images in an end-to-end fashion. We show how this algorithm works for both synthetic and real datasets. We observe that sequence-less models like deep convolutional neural network and U-Net show superior performance than sequence-based models like LSTMs and ConvLSTMs. For real images, we show performance of these networks on a digital retinopathy dataset (DRIVE). We also show how vessel thickness estimation can be incorporated into this image-to-tree approach. While results are preliminary, our exploration of visualization teachniques demonstrate the potential of our approach for novel ways of analysis.

References

Russell Bates, Benjamin Irving, Bostjan Markelc, Jakob Kaeppler, Ruth Muschel, Vicente Grau, and Julia A Schnabel. Extracting 3d vascular structures from microscopy images using convolutional recurrent networks. *arXiv preprint arXiv:1705.09597*, 2017.

Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

Brian E Chapman, Holly P Berty, and Stuart L Schulthies. Automated generation of directed graphs from vascular segmentations. *Journal of biomedical informatics*, 56:395–405, 2015.

Matthew J Czarny and Jon R Resar. Diagnosis and management of valvular aortic stenosis. *Clinical Medicine Insights: Cardiology*, 8:CMC–S15716, 2014.

John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. Graphvizopen source graph drawing tools. In *International Symposium on Graph Drawing*, pages 483–484. Springer, 2001.

Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R Rudnicka, Christopher G Owen, and Sarah A Barman. Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine*, 108(1):407–433, 2012.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Claudia Kondermann, Daniel Kondermann, and Michelle Yan. Blood vessel classification into arteries and veins in retinal images. In *Medical Imaging 2007: Image Processing*, volume 6512, page 651247. International Society for Optics and Photonics, 2007.

Louisa Lam, Seong-Whan Lee, and Ching Y Suen. Thinning methodologies-a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 14(9):869–885, 1992.

Chris McIntosh and Ghassan Hamarneh. Vessel crawlers: 3d physically-based deformable organisms for vasculature segmentation and analysis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1084–1091. IEEE, 2006.

Thanh T Nguyen and Tien Yin Wong. Retinal vascular changes and diabetic retinopathy. *Current diabetes reports*, 9(4):277–283, 2009.

Alireza Osareh, Majid Mirmehdi, Barry Thomas, and Richard Markham. Classification and localisation of diabetic-related eye disease. In *European Conference on Computer Vision*, pages 502–516. Springer, 2002.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Joes Staal, Michael D Abràmoff, Meindert Niemeijer, Max A Viergever, and Bram Van Ginneken. Ridge-based vessel segmentation in color images of the retina. *IEEE transactions on medical imaging*, 23(4):501–509, 2004.

SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

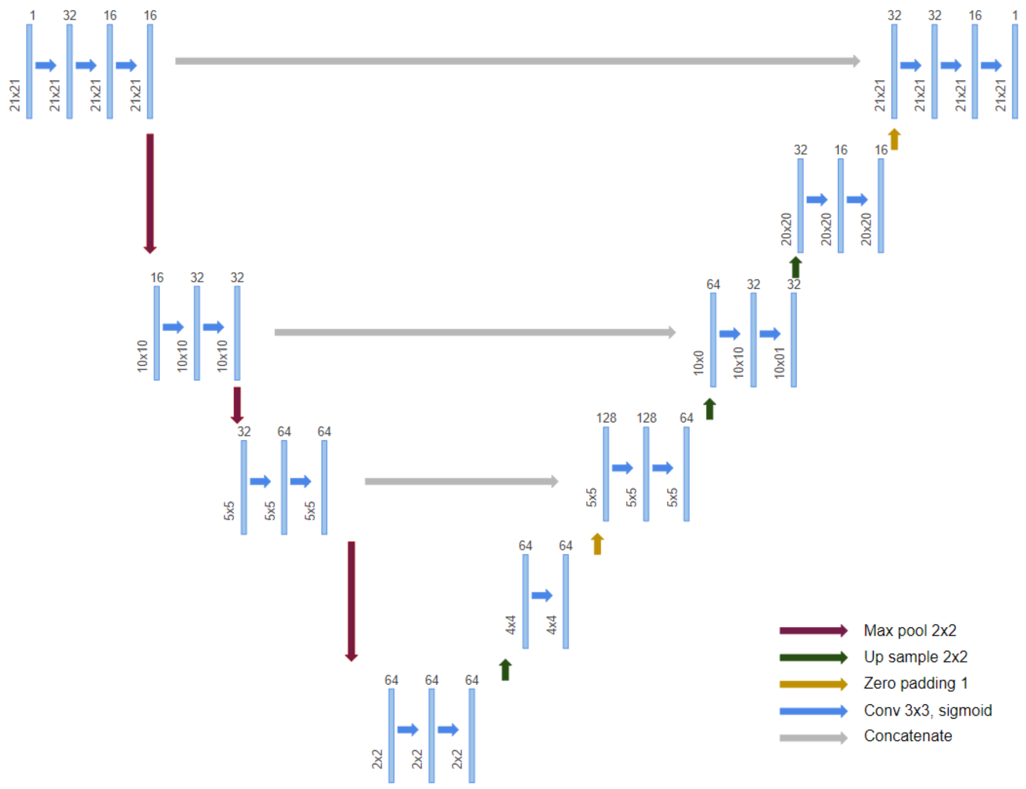


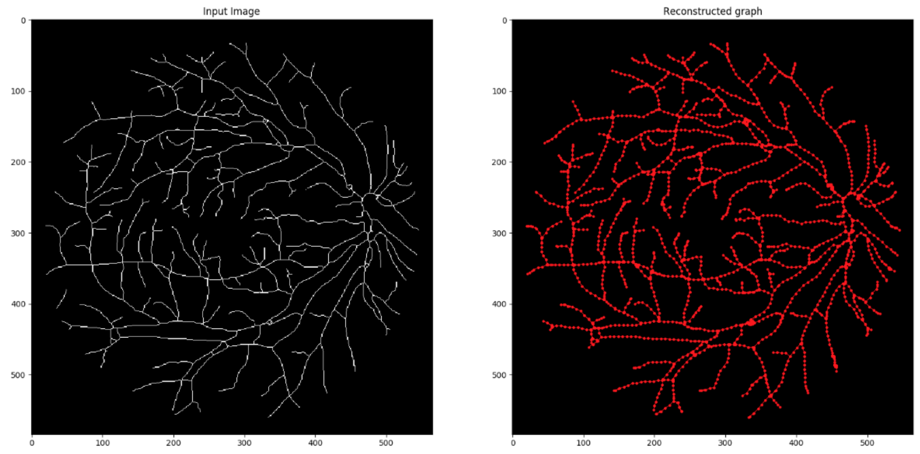
Figure 9: U-Net model used for training on real and synthetic patch data

Appendix A. Additional figures and results

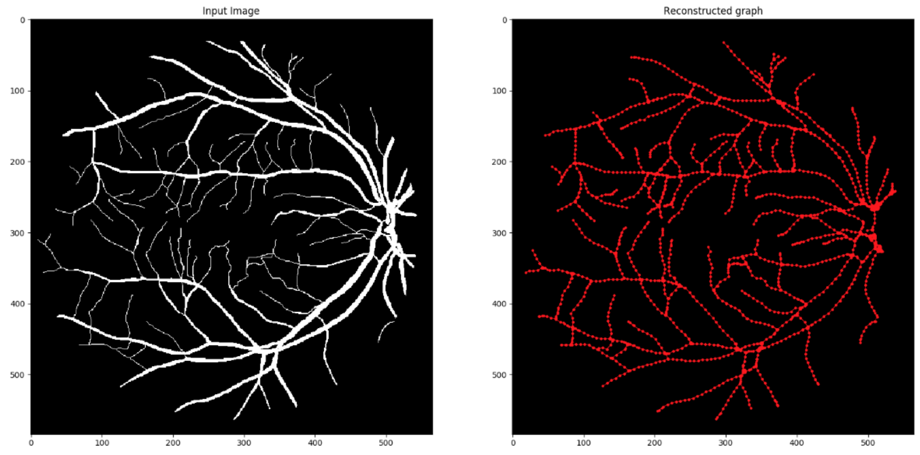
A.1. U-Net

U-Net used for tree construction is shown in Figure A.1. All convolution filters have size 3×3 and have sigmoid activation. Zero padding of input filter maps is applied wherever required.

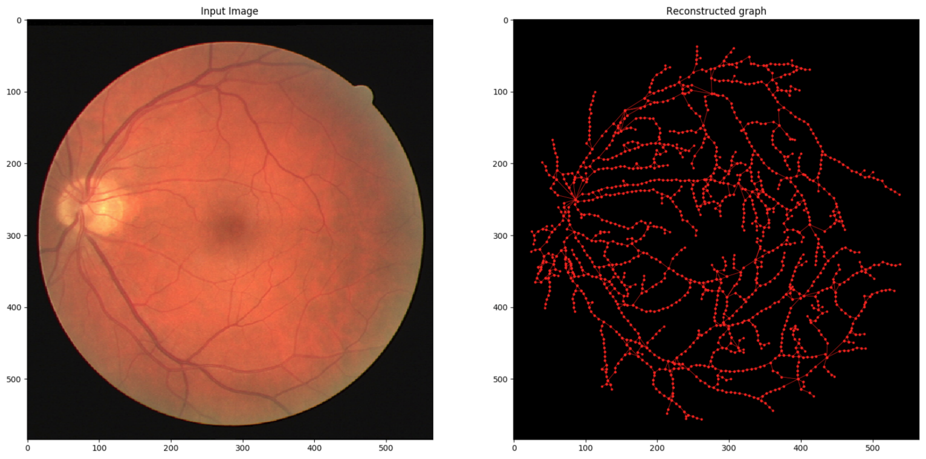
A.2. Reconstruction results for three levels



(a) Skeleton to tree



(b) Segmentation to tree



(c) Retinal image to tree

Figure 10: Reconstruction results