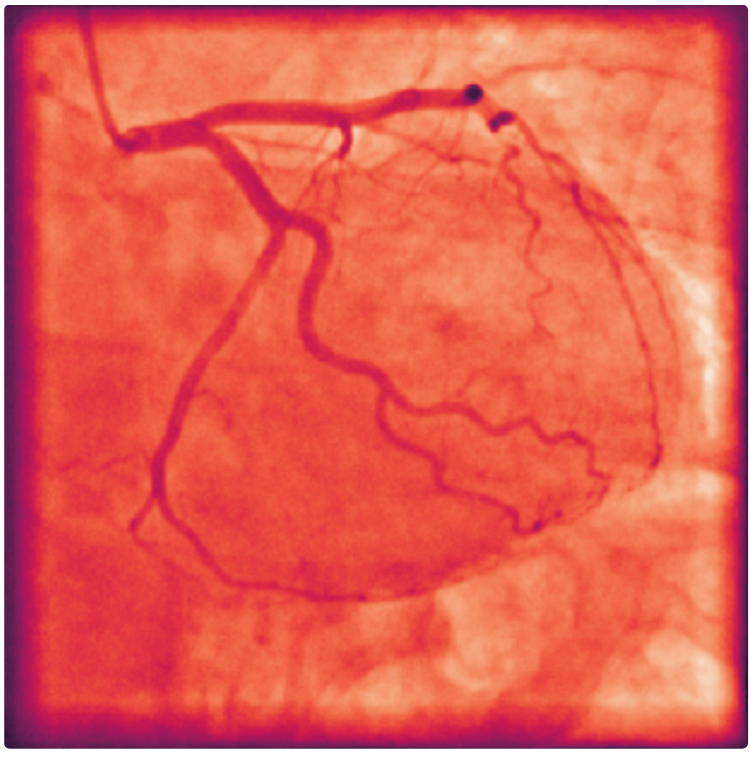
BD Progress report

Created by Katarzyna Heryan
Last updated: Feb 14, 2023 by Witold Serwatka • 6 min read

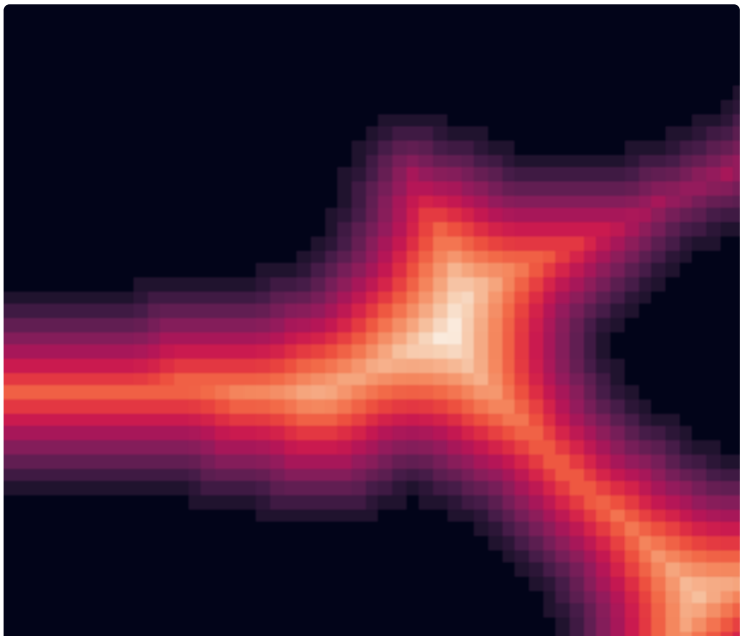
04.01.2023 - Intro

After initial analysis of existing solutions a basic design of the bifurcation detection algorithm was designed. The following explanation uses the below image:



As a first step, the image is segmented, resulting in a binary mask. This mask designates the pixels on which the euclidean distance transform in the following step will be calculated.

The euclidean distance transform (EDT) calculates for every pixel designated as a ‘1' the nearest euclidean distance to a pixel designated by a '0’. The result is the following image:



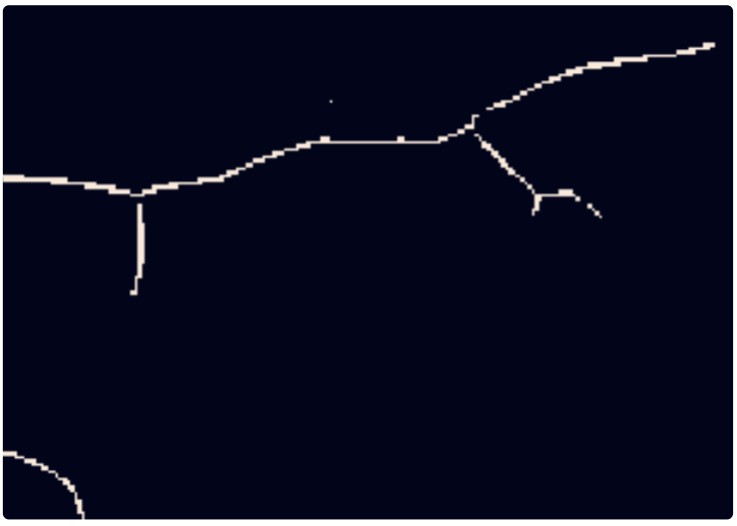
Enhanced image.

In this image the center of the vessel has the value of the radius of the vessel at any given point. This allows us to select a starting point for the centerline detection algorithm. The algorithm can be described in the following steps:

1. Select a point on the vessel ridge.
2. Create a circle with the radius equaling the value of the selected point.
3. Select points on the circumference of the circle that lie simultaneously on the vessel ridge.
4. Using these points, go to step 1.

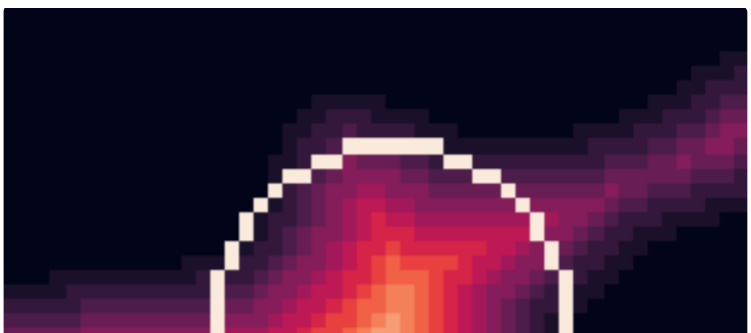
Intuitively, the best starting point would be the point where the vessel has the biggest radius.

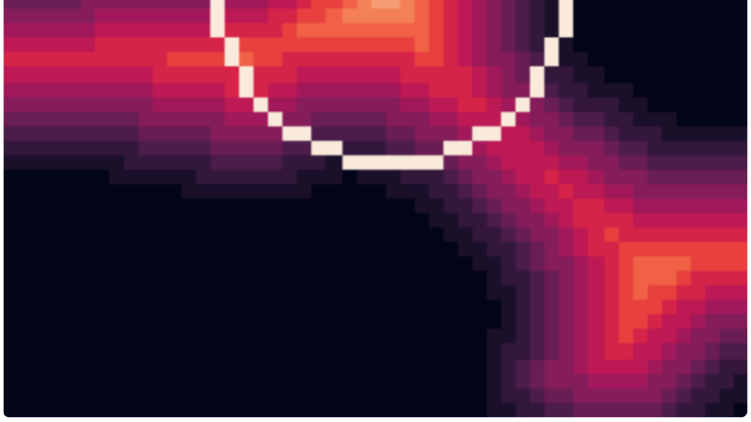
This algorithm requires us to select pixels belonging to the vessel ridge. This is done by calculating the gradient of the EDT image, as ridge pixels will have a gradient approaching 0. The gradient image is binarized using the thresholding method. Next by making use of the segmented image, we select only pixels with a small gradient that lie inside the vessel. This results in the following image:



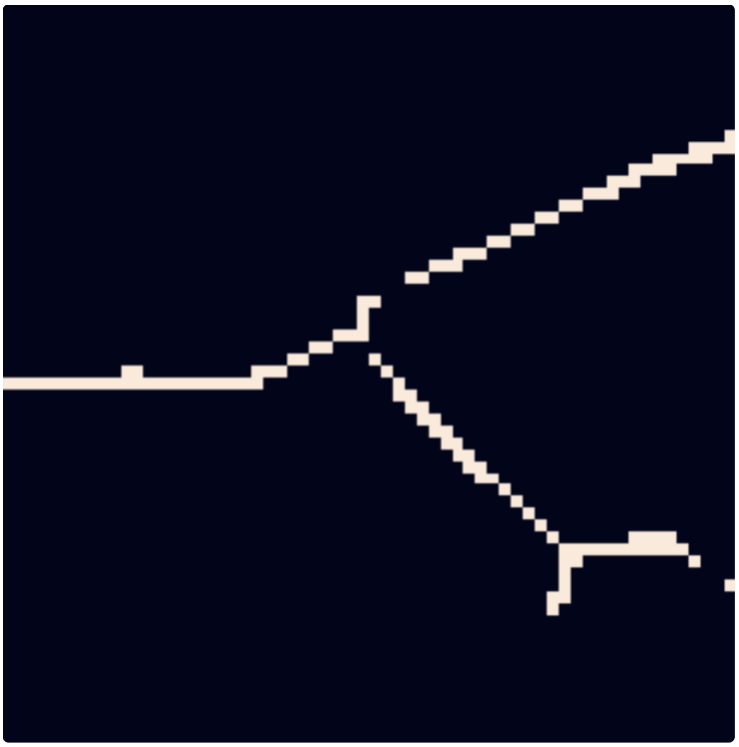
Enhanced image.

Based on this ridge image we look for pixels that simultaneously belong to the circle described above and the ridge. We simply select the intersection of these points.

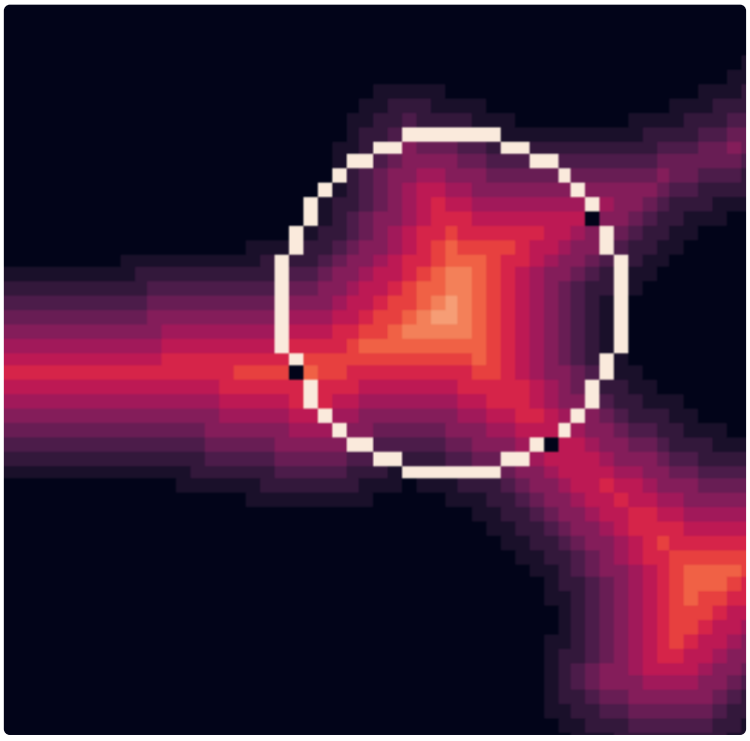




Circle pixels depicted in white.



Ridge pixels corresponding to the enhanced image on the left.

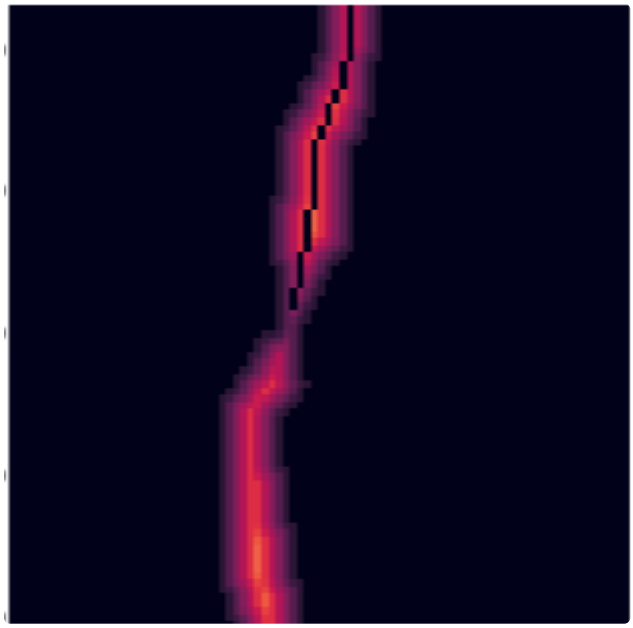


New points to analyze shown in black.

Based on this algorithm, we build a graph with the starting point as it's root and directed edges showing the 'direction' in which the graph expanded. By moving forward with the point selected as circle centers, we create an effect similar to rolling a ball along the vessel ridge. The resulting graph will have metadata associated with each node, useful for further analysis of the vessel structure. Bifurcation points will be detected as the points in the graph which have at least three connections.

23.01.2023

- Method was changed to detecting local maxima on circle instead of selecting from ridge pixels.
- Problem was detected: centreline detection algorithms stops on stenosis.

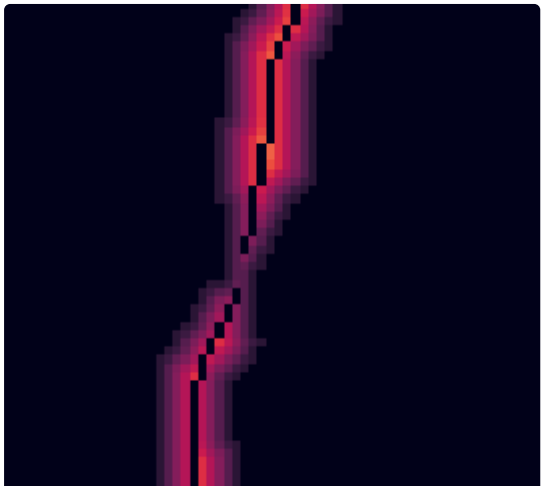


Detected centreline depicted in black.

- Solution: implement a starting point stack.

24.01.2023

- Implemented a starting point stack - potential starting points are selected using the peak_local_max function from the skimage library. They are sorted by vessel diameter, with the largest on top of the stack. It is checked whether they were already visited and if not, the graph is being generated from this point.

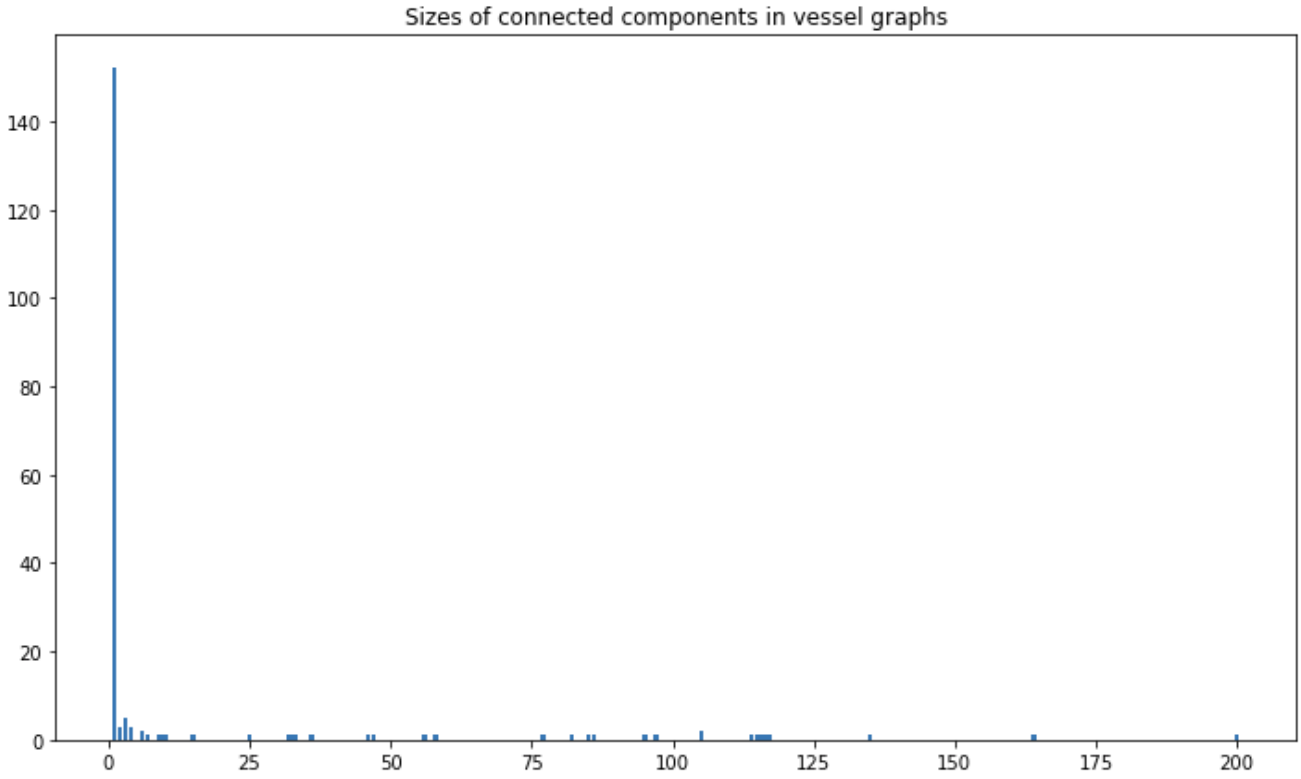


Detected centreline depicted in black, now being continued.

- Next step: Try to connect the disconnected centrelines.

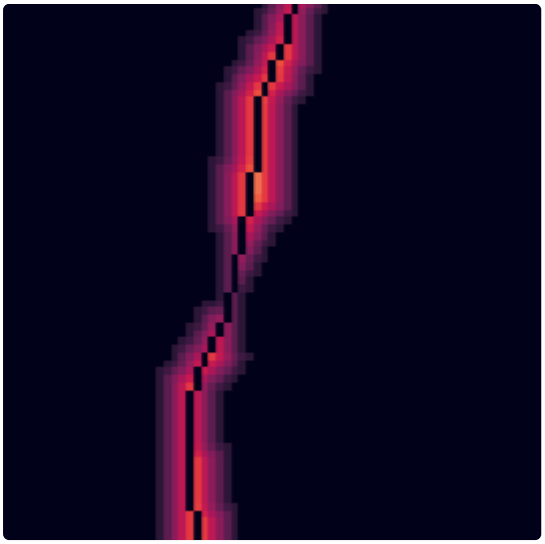
25.01.2023

- Implemented algorithm to remove small components of graph. Small components were defined as having less than 4 nodes based on the following diagram.

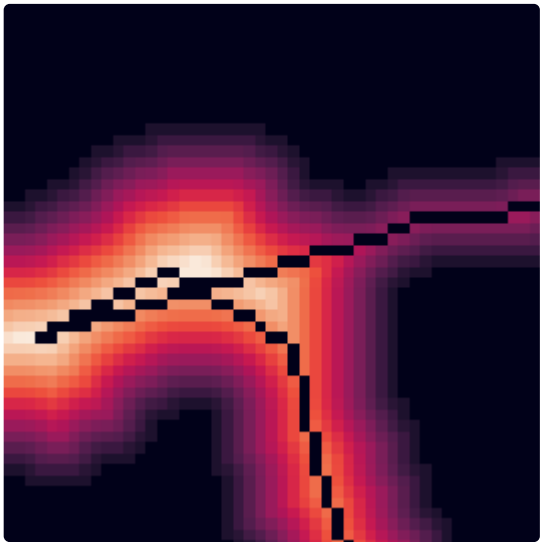


Based off 17 expert segmentations.

- Implemented basic algorithm to connect graph components. Results are depicted below:



Example of correct connection.

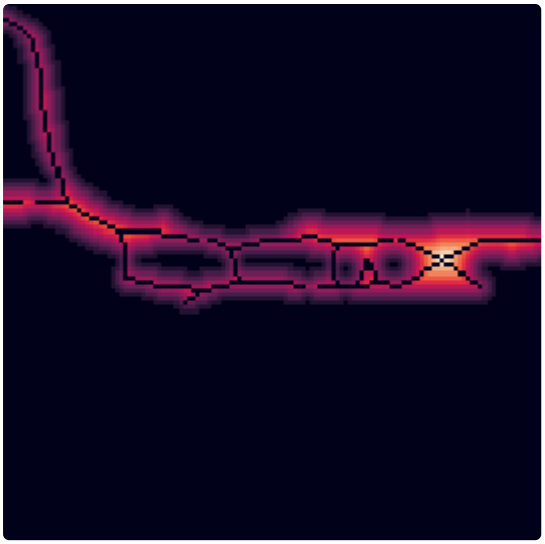


Result of erroneous connection.

- Idea to use BFS for connecting to nodes of degree higher than 1 in order to fix

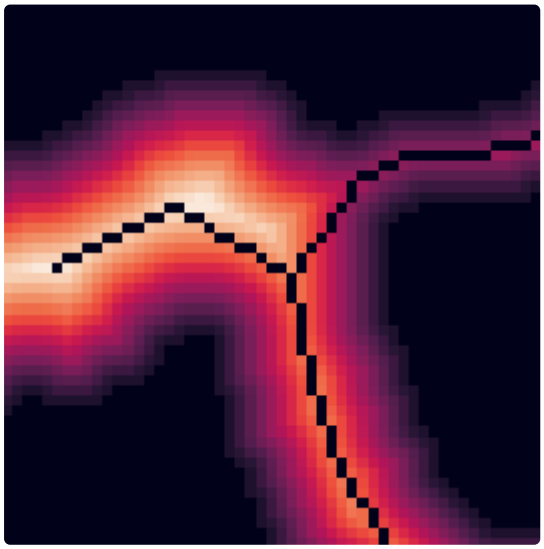
erroneous connections (right image).

- Problem detected: in overlapping vessels, the algorithm becomes unstable (see image below). According to MDs, these vessel fragments are not useful for diagnostic purposes, so we will ignore this issue for now. MDs should not select these fragments if they serve no diagnostic purpose.

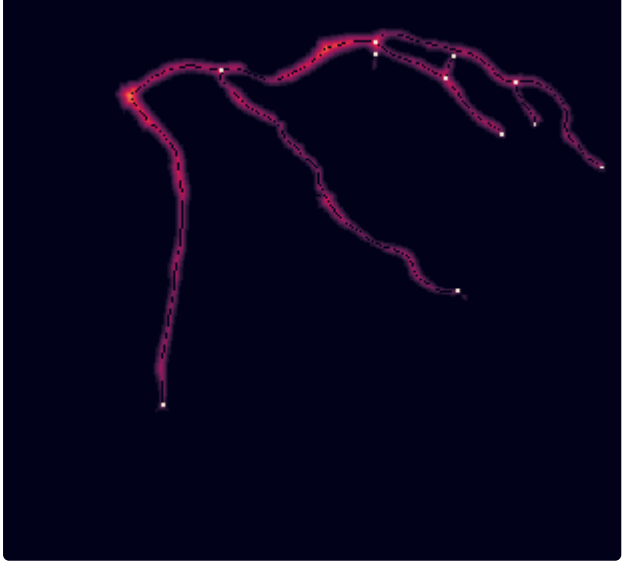


26.01.2023

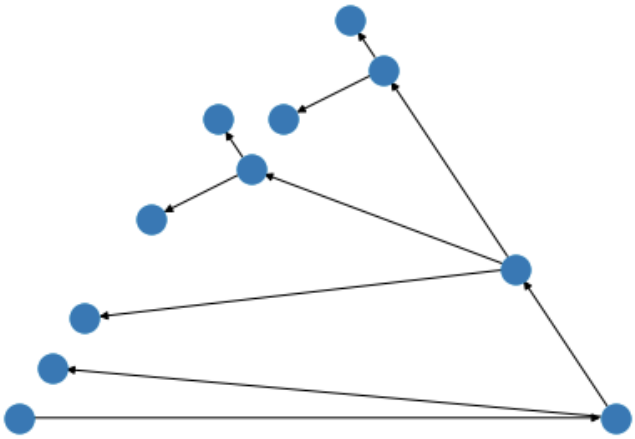
- Improved vessel connection algorithm by adding a BFS node search in the vicinity of an ending. The search only applies to nodes in the connected component of the segmentation the ending belongs to. The number of pixels visited correlates to the diameter of the vessel in the ending node. The result can be seen below:



- Basic version of bifurcation detection was implemented. The algorithm starts from a node with the largest vessel diameter, treating it as the root of the vessel tree. Next, it traverses the vessel graph along the available routes from the starting point. During this traversal, it will collect metadata on the vessel segment (width, etc., unimplemented yet). When it encounters a bifurcation or ending point, the collected data is summarized and saved as a node in a generalized vessel graph. The results and detected bifurcation/ending points can be seen below:



Detected bifurcation/end points depicted in white.



Generalized vessel graph.

09.02.2023

Problems to be considered:

- the graph generation function assumes there is only one component in the graph and only searches the one with the node containing the biggest vessel diameter
- the function also assumes the root is the node with the biggest vessel diameter

Ideas

- roi density estimation for proper root choice (comparison of densities between root candidates)
- MSC Konieczny for graph description

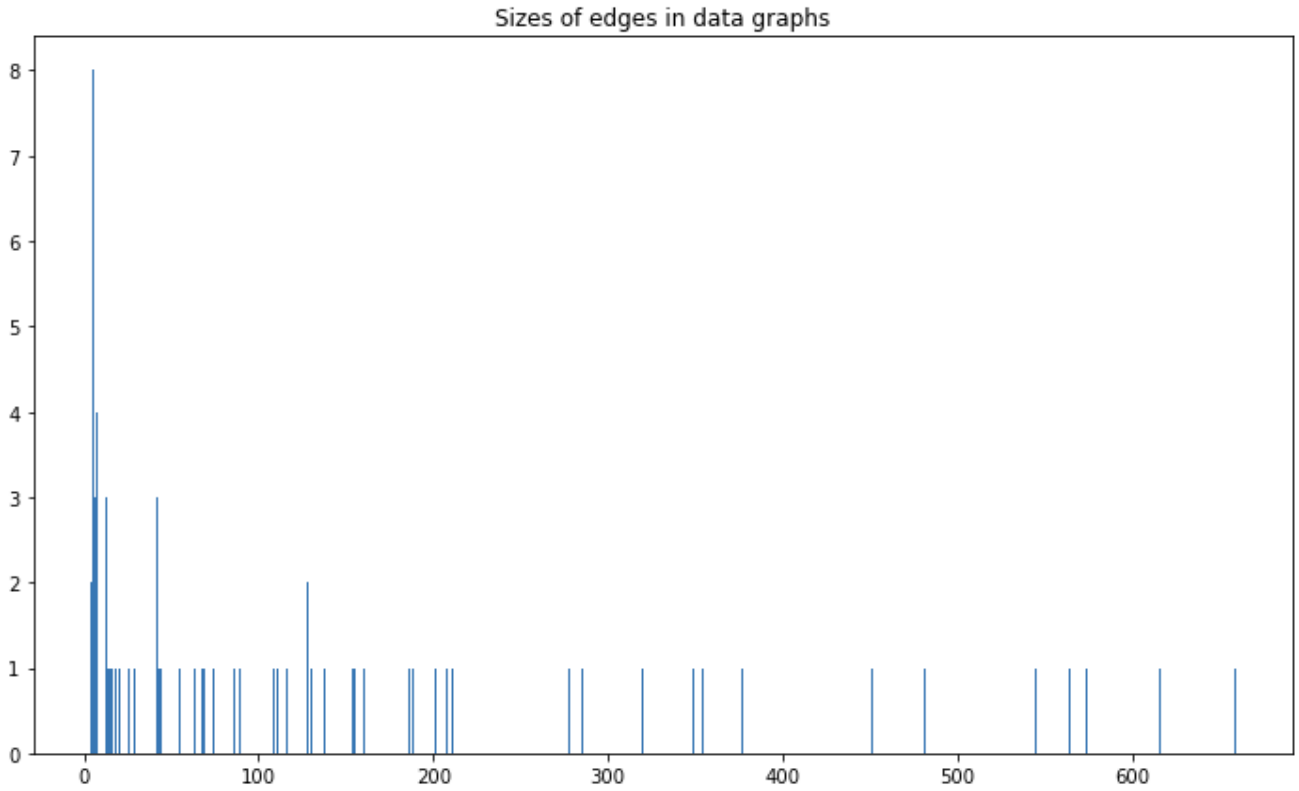
Next steps:

- data generation for segmentation, centreline, and bifurcation (documentation)
- bifurcation vol 1 done, graph generation for all cases + consider data for description (MSc)

- sota **stenosis**, small object detection
  - (skin lesions - ML, DL: feature, feature selection, classification, ...),
  - (retinal images) - vascular hypertension, lesions, retinopathy, diabetes
  - features for lesions detection
  - future: features for lesion description (skip measurements and regression for SyntaxScore) - data-dependent + augmentation, federated learning, transfer learning

14.02.2023

- Added data graph cleaning - if segment in data graph is too short, it is removed. Consequently, the nodes with degree 2 are removed and the edges originating in them are merged.



Based on this graph, edges shorter than 10 are removed.

- Changed the way the root node is chosen - it is chosen from the two nodes of the segment with the biggest average vessel diameter. If one of the nodes has degree 1, then it is selected. Otherwise, the node with the bigger vessel diameter is selected. This method chooses the correct root 14/17 times (82%).

Correct root selection.

Wrong node selection.

+ Add label

Be the first to add a reaction