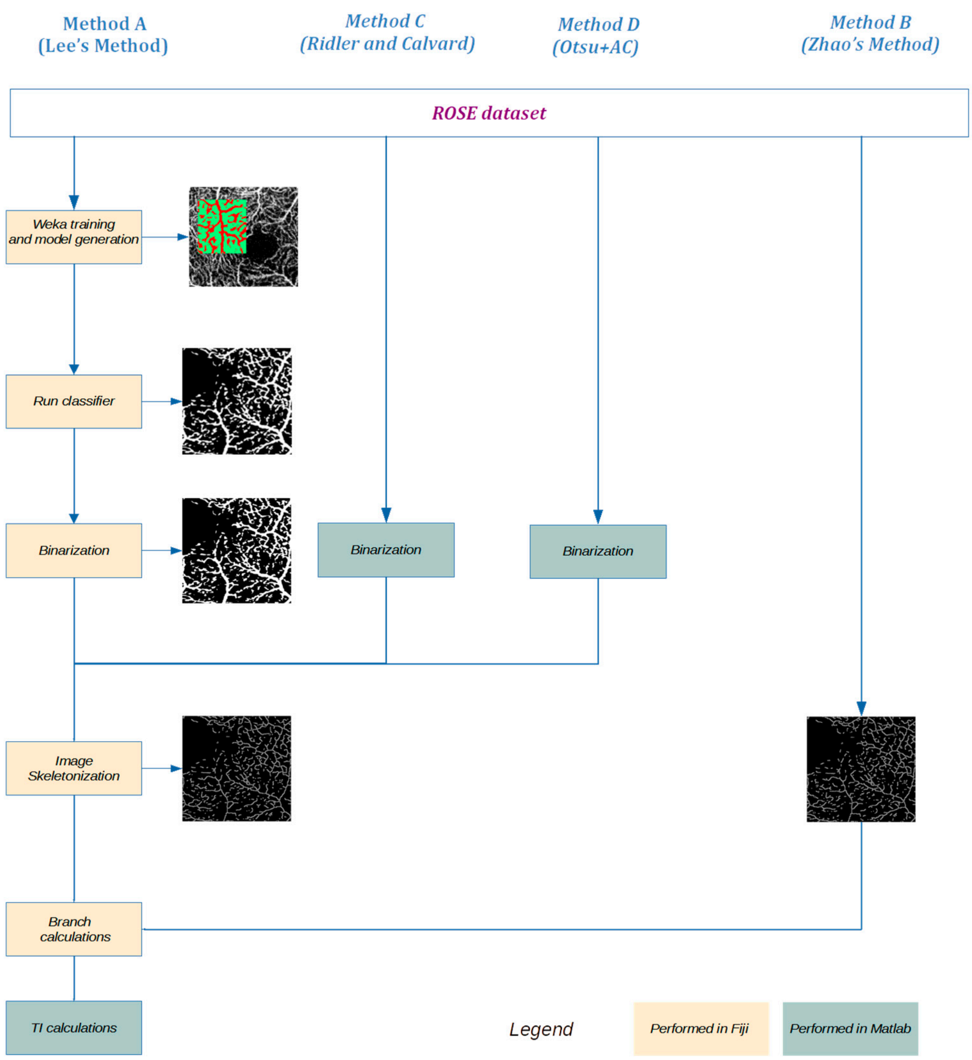
Supplementary S1

Input images

For each of the considered methods, images 11_OS_SVP (normal) and 27_OD_SVP (tortuous) from the ROSE-2 dataset, original test image were chosen. In the ROSE dataset, they can be retrieved with the following path:
ROSE\data\ROSE-2\test\original\11_OS_SVP.png
ROSE\data\ROSE-2\test\original\27_OD_SVP.png

Image Processing and branch information calculations workflow

All image processing was performed using a combination of Mathwork's Matlab [38], Fiji [39] and the Trainable Weka Segmentation Fiji plugins [41].
The whole processing flow can be visually described as in the following figure.



Detailed steps needed to replicate the methods are reported in the following.

Method A (Lee's Method)

Weka Model Training

A representative OCTA image from the ROSE-1 dataset was used to train the software to classify vessels from the background by deriving foreground and background information from the ROSE segmentation (superficial OCTAs, train folder, image 18).
Detailed steps necessary to reproduce the process:

1. open in Fiji the binarized image;
2. set image foreground to 1 and background to 0 (Process-math-multiply (*255), Edit-Invert);
3. select the foreground (edit-selection-create);
4. add the selected foreground to the ROI manager (edit-selection- add to manager);
5. select a 128 pixel wide upper left corner of the image (Edit-selection- specify :width =128, height=129, x coordinate=0,ycoordinate=0) and add it to the ROI manager (edit-selection- add to manager);
6. from the ROI manager, create the foreground for the upper left corner (ROI manager: select the two previously added rois, more- AND). Remove from the manager the roi for the whole image. Save the foreground roi (ROI manager- more- save);
7. To create the corresponding background roi execute the following steps:
8. edit-invert (both foreground and background);
9. edit-selection-create (selects the background only)
10. edit-selection- add to manager. Repeat step 6 for the whole image;
11. save the background roi;
12. Launch the Weka segmentation plugin (segmentation-trainable Weka segmentation)
13. Reopen the ROI manager, select the foreground, and add the roi as Weka Class 1. Repeat for the background (add as Class 2)
14. Select the training settings (select Gaussian blur, Sobel filter, Hessian, Difference of Gaussian, Membrane projections - membrane thickness 1, membrane patch size 19, Minimum sigma 1, Maximum sigma 16);
15. Train the classifier and save it for subsequent use.

Image binarization using the Weka ML model

In order to apply the model:

1. Load the image in Fiji;
2. Open the Weka plugin;
3. Load the classifier;
4. Get probability.

Those steps result in a two slice probability map. In order to have a binarized image:

1. Select the second slice (the one with white background: image- stacks- set slice 2);
2. Remove the second slice (image- stacks- delete slice);
3. Convert the image to 16 bit (Image-type-16 bit);
4. Binarize the image (Image- adjust- autothreshold - max entropy, white objects on black background);
5. Save the binarized image.

Binarized image skeletonization, branch information and tortuosity index calculations were performed using the steps described in the Common procedures for all methods section below.

Method B (Zhao's Method)

Normal and Tortuos skeletonized images can be retrieved directly from the Rose dataset with the following path:
ROSE\data\ROSE-2\test\gt\11_OS_SVP.png
ROSE\data\ROSE-2\test\gt\27_OD_SVP.png

Branch information and tortuosity index calculations were performed using the steps described in the Common procedures for all methods section below.

Method C (Ridler and Calvard)

Image binarization was performed in Matlab using established techniques: first, raw images histograms were equalized using Niblack's algorithm, histogram equalization) followed by image binarization by the method of Ridler and Calvard. The implementation, while extremely simple, uses several Matlab scripts that are available from the authors at reasonable request.

Binarized image skeletonization, branch information and tortuosity index calculations were performed using the steps described in the Common procedures for all methods (section below).

Method D (Otsu + active contour)

Image binarization was performed by a simple script implementing Otsu's an Chan's method, using Matlab's built-in functions.

```
I= imread(filename);
level = graythresh(I);
mask = im2bw(I,level);
% Filter components by area
BW = bwareafilt(mask, [197 Inf]);
% Evolve segmentation
BW = activecontour(I, BW, 100, 'Chan-Vese');
% Form masked image from input image and segmented image.
Binarized = I;
Binarized (~BW) = 0;
```

Binarized image skeletonization, branch information and tortuosity index calculations were performed using the steps described in the Common procedures for all methods (next section).

Common procedures for all methods

Binarized image skeletonization

Using Fiji:

1. Load the binarized image;
2. Plugins skeleton-skeletonize (2d-3d);
3. Save the skeletonized image.

Branch information calculations

Using Fiji:

1. Load the skeletonized image;
2. Analyze-skeleton –analyze skeleton (no pruning , show detailed info);
3. Save the skeletonization results (branch information in a csv file- file –save as).

TI calculation

TI can be calculated in Matlab importing the branch information csv files and summing up the Branch length and Euclidean distance columns with simple scripts implementing Eq 1.

$$TI = \frac{\sum Branchlengths}{\sum Euclideandistances} \qquad \text{Eq. 1}$$

As an alternative, the same computations can be performed loading the branch information csv file in a spreadsheet. Numerator and denominator of Eq.1 are computed simply adding up Branch length and Euclidean distance columns of the sheet.