IA716 - Computer Vision

Vessel segmentation on SLO images

Link to the GitHub repository

Realized by:
Paul Aristidou
Olivier Lapabe-Goastat

Teacher :
Gianni Franchi

February 29, 2024

Contents

1 Introduction

The primary objective of this project is the development and implementation of advanced image segmentation techniques specifically focused on the precise identification of retinal blood vessels in ocular imagery. Unlike conventional approaches that heavily rely on deep learning and other AI methodologies, this project adopts a "classical" approach, leveraging morphological image processing techniques. This choice was made intentionally to provide a fundamental understanding and hands-on experience with traditional image segmentation methods, thus offering a "ground-up" learning perspective in the field of computer vision.

In this report, we will detail the various methodologies we tried and applied, emphasizing the non-AI approach to morphological image processing. We will discuss the optimization of segmentation parameters, assess the effectiveness of various morphological operations, and evaluate our results. Our goal is to not only validate the efficacy of traditional image segmentation methods in medical applications but also to provide an educational framework for understanding these techniques' underlying principles and applications.

2 Presentation of the project

2.1 Objective

The primary objective is to automate the segmentation of retinal blood vessels from the given images. The evaluation function employs morphological skeletonization to measure the precision, recall, and F1 score between the segmented output and the ground truth, providing a quantitative assessment of our segmentation accuracy.

2.2 Approach

For this project, we followed the following approach / methodology:

- Initial analysis of raw images and literature review to identify potential image processing strategies, drawing inspiration from key studies (see bibliography).
- Development of a transformation pipeline to enhance image features and facilitate accurate segmentation, including contrast enhancement and morphological operations.
- Optimization of parameters to refine segmentation results.

3 Transformation steps for segmentation

Throughout the development of our segmentation approach, various methodologies were explored and assessed for their effectiveness in isolating retinal blood vessels. After extensive experimentation, the following pipeline emerged as the most effective, balancing detail preservation with noise reduction and artifact removal:

1. Contrast enhancement (*contrast_factor*): Adjusted to improve the visibility of retinal vessels against the surrounding tissue, set by the parameter *contrast_factor*.

2. Opening by structures (*p_structure*): This step was crucial for preprocessing the retinal images. By applying morphological opening with a variety of structuring elements (*p_structure*), we were able to effectively remove noise and small artifacts that could be mistaken for blood vessels, while simultaneously smoothing the outlines of actual vessels. This not only cleaned the image but also helped in making the vessel structures more pronounced for subsequent processing steps.

3. Black tophats by structures (*p_structure*): Given the varying contrast and visibility of retinal vessels, black tophat operations were employed to enhance dark vessels against the lighter retinal background. This step, parameterized by the same array of structuring elements (*p_structure*), was instrumental in highlighting vessels of different sizes and ensuring that even faintly visible vessels were accentuated, facilitating their detection and segmentation.

4. Erosion with reconstruction (*p_structure*): To further refine the segmentation, erosion followed by morphological reconstruction was utilized. This process effectively simplified the vessel structures to their core forms, removing extraneous pixels that did not contribute to the primary vessel pathway. Subsequent reconstruction with the original image as a guide ensured that the essential structure of the vessels was preserved while maintaining their true scale, an essential step for accurate segmentation.

5. Gaussian filtering (*sigma*): Image smoothing via Gaussian filtering, controlled by the *sigma* parameter, was a key step in reducing image noise without blurring the critical edges of blood vessels. This delicate balance was vital for maintaining the integrity of vessel borders while preparing the image for effective thresholding.

6. Thresholding (*threshold*): The application of thresholding transformed the preprocessed image into a binary representation, crucial for the final segmentation. This step ensured that only structures meeting the intensity criteria (above the *threshold*) were preserved, effectively isolating the vessels from the background.

7. Remove small objects (*min_size*): Post-segmentation, it was imperative to remove remaining noise and non-vessel artifacts. Setting a *min_size* parameter allowed for the exclusion of small, irrelevant objects, ensuring that the segmented output closely represented the actual vascular network without contamination from spurious elements. The challenge of this operation was to find the right balance in the size of the objects we wanted to remove, trying to keep as much information as possible and removing all the extra noise.

8. Median filtering by structures ($p\_structure$): As a concluding post-processing step, median filtering was applied using the same structuring elements ($p\_structure$). This step was instrumental in smoothing the segmented vessels, reducing pixelation and artifacts, and further refining the segmentation output for a cleaner, more clinically useful representation of the retinal vasculature.

This segmentation pipeline was meticulously refined through iterative testing, with each step rigorously evaluated with the provided evaluation function. The parameters were optimized based on a comprehensive evaluation metric, incorporating precision, recall, and the F1 score, leading to the development of our optimized segmentation pathway.

4 Parameters tuning

Now that we have found the transformations we want to apply, the immediate next step is to finetune the parameters of the different operations that we are applying to the images.

As introduced in the previous part, we have 5 parameters to finetune : *contrast_factor*, *sigma*, *threshold*, *min_size* and *p_structure*. (To allow for consistency across the operations, we will take the assumptions of having the same *p_structure* for the relevant operations.)

For this, we run a grid search to test different values of our parameters, using the ProcessPoolExecutor() executor to enable using a pool of worker processes.

4.1 Narrowing down breadth of possible values

First, we narrow down the breadth of possible values for each. We have first played "by hand" to narrow the bandwidth of the possible parameters :

- *contrast_factor*: after trying values uniformly spread between 0 and 5, we have identified that the best value sits between 1 and 2. We will then run a grid search within the list: range(10, 20)/10.
- *sigma*: after trying values uniformly spread between 0 and 5, we have identified that the best value sits between 0.5 and 1.5. We will then run a grid search within the list: range(5, 15)/10.
- *threshold*: after trying values uniformly spread between 0 and 50, we have identified that the best value sits between 5 and 15. We will then run a grid search within the list: range(5, 15).
- *min_size*: after trying values uniformly spread between 0 and 500, we have identified that the best value sits between 50 and 100. We will then run a grid search within the list: range(50, 100, 5).
- *p_structure*: this parameter is the most difficult to play with, as it comprehends several sub-parameters: the shape (e.g., disk, rectangle), the dimensions of the shape (width, breadth) and the rotation angle of the shape. **This will allow multi-scaling and multi-orientation in the different morphological operations.** After trying different combinations of shapes (disk, rectangle and diamond), we have identified that disk shapes between sizes 2 and 4 are good footprint shapes to start with [[disk(1), disk(2)], [disk(1), disk(2), disk(3)], [disk(1), disk(2), disk(3), disk(4)]], then expanding to all other combinations (see methodology below).

4.2 Gridsearch in 3 steps

Running one grid search on all parameters will be a challenge, especially with *p_structure* for which the number of possible combinations is massive (shape, dimensions and rotation angle). Therefore, we will run the grid search in 3 steps, having the **f1score averaged**

across all images as evaluator.

The first step goes a first grid search with the possible ranges of parameters identified "by hand" in section 5.1.

Parameter	**Range to test**
contrast_factor	range(10, 20)/10
sigma	range(5, 15)/10
threshold	range(5, 15)
min_size	range(50, 100, 5)
p_structure	[[disk(1), disk(2)], [disk(1), disk(2), disk(3)], [disk(1), disk(2), disk(3), disk(4)]]

Table 1: Gridsearch - First step

After the first run, we find optimal value for the parameters *contrast_factor*, *sigma*, *threshold*, *min_size* and we fix them for the second gridsearch run. Playing with different shapes, sizes (for multi-scaling) and rotation angle (for multi-orientation), rectangles seem the best *p_structure* possibilities. The second run will operate a grid search on all possible p_structure rectangles:

Sub-parameter for *p_structure* rectangles	**Range to test**
number of combined rectangles	[1, 2, 3]
width	[2, 3, 4]
height	[1, 2, 3]
rotation angle in degrees	[0, 45, 90]

Table 2: Gridsearch - Second step

We find an optimum for [rect(3,3,0), rect(3,3,45), rect(3,3,90)] Ffor the third and last run, we fix this array of *p_structures* and operate a grid search on the other parameters :

Parameter	**Range to test**
contrast_factor	range(10, 20)/10
sigma	range(5, 15)/10
threshold	range(5, 15)
min_size	range(50, 100, 5)

Table 3: Gridsearch - Third step

We then run the grid search on all possible combinations of parameters, returning 3 evaluation indicators, averaged on all images: *precision*, *recall* and *f1-score*. The optimal set of parameters corresponds to the best *f1-score*. We use the ProcessPoolExecutor() executor, enabling us to use a pool of worker processes.

Finally, the last step of the grid search returns the optimal following set of parameters :

Parameter	**Optimal Value**
contrast_factor	1.5
sigma	0.9
threshold	10
min_size	75
p_structure	[rect(3,3,0), rect(3,3,45), rect(3,3,90)]

Table 4: Optimal parameter values for the algorithm.

These best parameters help achieving these results (averaged on all images):

Evaluation indicator (mean on all images)	**Value**
Best *f1-score*	**77.46%**
Corresponding *precision*	82.98%
Corresponding *recall*	73.30%

Table 5: Table showing the three values.

5 Results

With the optimal set of parameters identified, let's now have a deeper look at the results on *precision*, *recall* and *f1-score*. The following table displays each of the evaluation indicators on each image. Each image and its segmentation can be found in the */data/results* folder in our github repository (link).

Image	**f1-score**	**precision**	**recall**
Image 01	77.03%	79.14%	75.03%
Image 02	76.99%	69.62%	86.11%
Image 03	77.68%	89.06%	68.88%
Image 08	79.20%	94.28%	68.28%
Image 21	72.29%	80.43%	65.65%
Image 26	75.28%	83.13%	68.78%
Image 28	75.20%	82.17%	69.32%
Image 32	78.42%	80.51%	76.43%
Image 37	79.50%	83.08%	76.22%
Image 48	83.05%	88.36%	78.34%
Average	**77.46%**	**82.98%**	**73.30%**

Table 6: Results with optimal set of parameters

We will have now a deeper look at the images with the best and worst values in precision or recall and try to understand the reasons.

Let's start with the image with the highest f1-score (image 48), so that we can then compare it to the ones with the lowest precision (image 02) and lowest recall (image 21).

- Image 48 (best f1-score):

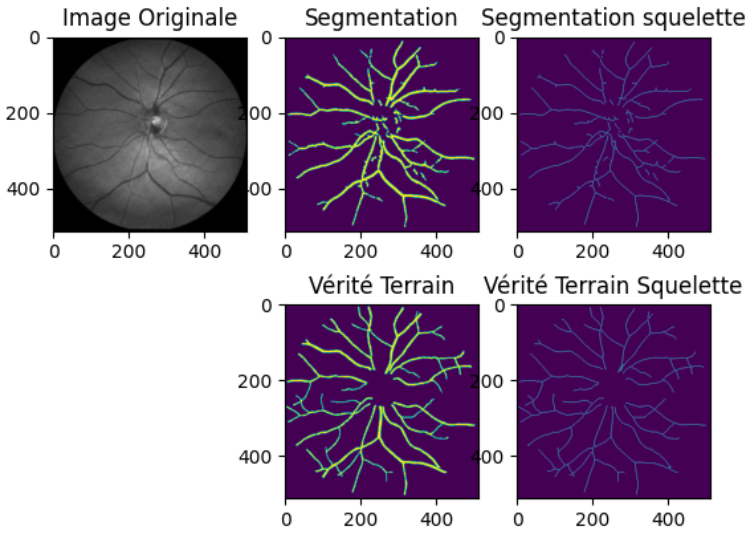

Figure 1: Segmentation results Image 48

- Image 02 (lowest precision):
  Image 02 has the lowest precision of all images. It means that there is a higher rate of False Positives. The figure below showcases it: the segmentation detects more vessels as in the Ground Truth. On the contrary, it has a quite good recall. Let's try to explain it and compare ground truth segmentations between image 48 and image 02.

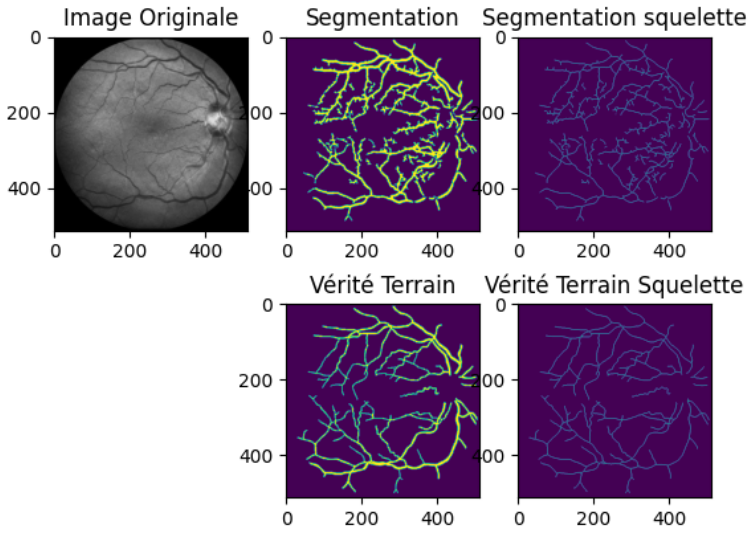

Figure 2: Segmentation results Image 02

- Image 21 (lowest recall):
  Image 21 has the lowest recall of all images. It means that there is a higher rate of False Negatives. The figure below showcases it: the segmentation detects fewer vessels as in the Ground Truth. On the contrary, it has a quite good precision. It is quite the opposite of image 02 where the rate of False Positives was higher.

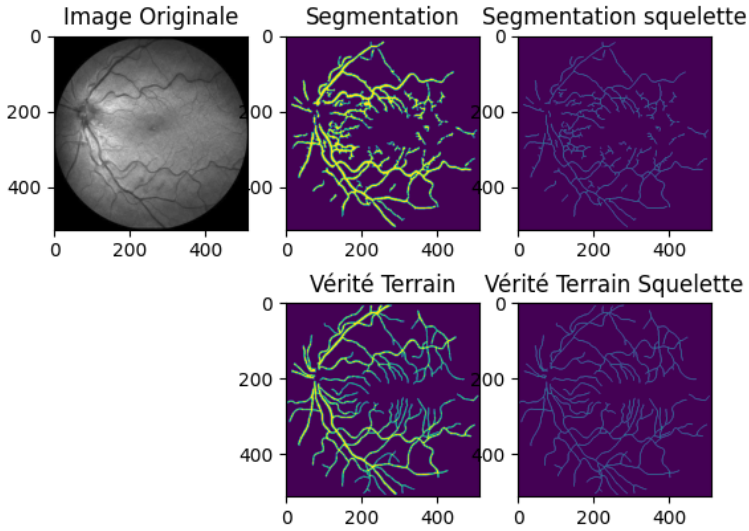

Figure 3: Segmentation results Image 21

By comparing these 3 images, 2 hypothesis emerge which could explain the low precision on image 02 and the low recall on image 21.

- Hypothesis 1 : Image 02 has a low variety of vein thicknesses compared to image 21 (especially looking at the middle of the picture). It is therefore more difficult on image 02 to distinguish a vein from a "non-vein", explaining the low precision. On the opposite, image 21 has a high variety of vein thicknesses (some very thick and a multitude of very small ones), which makes more difficult to identify the smaller veins, explaining the low recall.

- Hypothesis 2 : Image 21 has a higher light exposure compared to image 02. It makes it difficult to identify smaller veins on image 21. On the contrary, the algorithm identifies veins "too much" which creates a high False Positive rate.

Two morphogical operations might be at the source of these differences : the removal of small object (step 7) having an impact on our hypothesis 1, and the contrast enhancement (step 1) having an impact on our hypothesis 2.

However, we have optimized the parameters of our morphological operations to maximize the average f1-score on all images. The parameter finetuning helped to segment as good as possible all images, finding the right balance of our parameters, images 02 and 21 being the 2 extremes of this "right balance".

6 Conclusion

By using purely conventional methods without any deep learning, we could achieve reasonable vessel segmentation scores, taking into account the difference is scale and orientation of the veins. However, the results in recall and precision can vary quite a lot from one picture to another, which could be explained by the structural differences of the eye (variety of thicknesses of veins) and / or of the picture (e.g., light exposure). This is due to the optimization of the parameters, finding the right balance to optimize the f1-score averaged on all images of our training dataset.

To improve further the algorithm, one might want to implement the following ideas :

- The parameters were finetuned in a 3 step approach, due to the high number of potential combinations, especially due to the *p_structure* parameter. One idea might be to use high-performing parameter optimization algorithms to find a better calibrated set of optimal parameters.
- We faced some difficulties to identify the continuity of veins and avoiding the segmentation algorithm to "break" veins, which would then eventually disappear at the latest stages (e.g., remove small object). Keeping better this continuity of veins would be something to further investigate.

7 Bibliography

1. Xu, J., Ishikawa, H., Wollstein, G., & Schuman, J. S. (2008). Retinal vessel segmentation on SLO image. Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2258-2261.

2. Zana, F., & Klein, J.-C. (2001). Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. IEEE Transactions on Image Processing, 10(7), 1010–1019.

3. Ramos-Soto, O., Rodríguez-Esparza, E., Balderas-Mata, S. E., Oliva, D., Hassanien, A. E., Meleppat, R. K., & Zawadzki, R. J. (2021). An efficient retinal blood vessel segmentation in eye fundus images by using optimized top-hat and homomorphic filtering. Computer Methods and Programs in Biomedicine, 201(105949).