Politecnico di Torino

Master's degree in Biomedical Engineering
Medical Image Processing *– A.Y. 2025/2026*
Retinal Vessel Challenge

D'Aiello Laura, s346094
Darino Alessandra, s346001
Dona Camilla, s338964
Gurra Michele, s341858

1. Introduction

The retina is the only site in the human body where blood vessels can be directly visualized in vivo and, for this reason, it allows for a non-invasive visualization of the microvascular system, reflecting systemic vascular dysregulation and enabling early detection of both ocular and systemic diseases [1].

Characteristic retinal vascular changes, including arteriolar narrowing, venular widening, increased tortuosity and rarefaction show strong associations with major cardiovascular and systemic conditions such as hypertension, atherosclerosis, and diabetes mellitus [1].

The retinal vasculature plays a critical role also in ocular diseases such as glaucoma, age-related macular degeneration (AMD) and diabetic retinopathy [2].

Deep learning (DL) techniques have demonstrated diagnostic accuracy in retinal vessel segmentation that is comparable to human experts. DL-based automated systems can significantly reduce manual workload, accelerate segmentation process and lower operational costs. Their performance is clinically acceptable and helps minimize both intra-operator and inter-operator variability. Integrating these tools into real-world screening programs could enhance efficiency and substantially transform the diagnosis and management of retinal diseases in the near future [3].

The goal of this project is to develop an algorithm capable of automatically segmenting retinal vessels in color fundus photography images. The following sections will discuss the choices that led to the final strategy adopted. *Figure 1* illustrates an overview of the training and evaluation pipelines for vessel segmentation using a U-Net model.

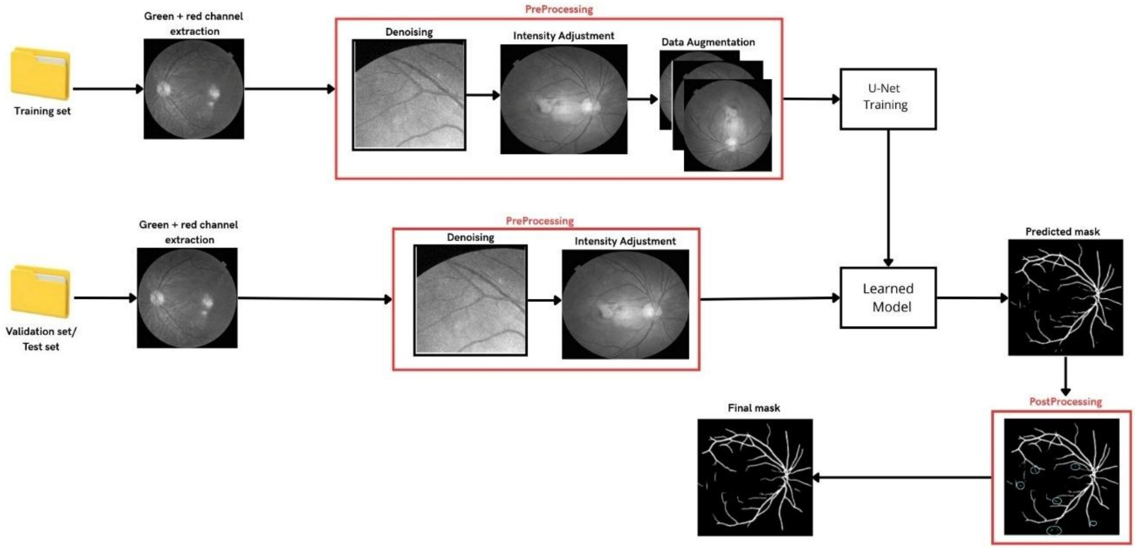

Figure 1: Schematic workflow of the proposed retinal vessel segmentation pipeline based on a U-Net architecture

2. Materials and methods

This section details the methodology used to develop the segmentation model, beginning with the initial dataset analysis and the definition of a baseline. Subsequently, it focuses on optimizing the pipeline through specific pre-processing techniques, hyperparameter tuning and post-processing refinements.

2.1 Dataset

The dataset used in this study consists of 600 images. The images are provided in RGB format, saved as PNG files and encoded in uint8 format. Each image is associated with a corresponding mask containing a manual segmentation performed by an expert operator. These manual segmentations were exported using the same format and encoding as the original images, with a binary representation in which the value 0 denotes the background and the value 255 indicates manually segmented vessels.

In addition to the original manual masks, the dataset includes a binarized version of the masks called manual_py, in which pixel values are 0 or 1. This representation was adopted to ensure compatibility with the input format expected by the neural network. The manual_py masks are equivalent to the original reference masks in terms of content but are formatted to be compatible with model training. All images and their corresponding masks have a spatial resolution of 1024×1024 pixels.

2.1.1 Dataset management

Initially, the provided division was maintained, with a training set of 500 images and a validation set of 100 images. During data cleaning, three images were removed from the training set.

- Image 039480. It featured a manual mask that didn't match the original image. Since the error was not correctable, it was excluded to avoid introducing label noise, which would have compromised the model's learning.
- Images 827255 and 277582. Both were extremely dark, making it impossible to distinguish the vascular structure and their corresponding manual masks were completely black. These cases were interpreted as acquisition errors and removed, as they did not contain useful information for segmentation and could have led the model to learn non-physiological patterns. Since these images represent two outliers out of a total of 600 images, their removal did not alter the distribution of the dataset. For this reason, they were not included in the validation set or in a possible "special" validation set, as they do not represent rare clinical cases, but simply incorrect acquisitions.

Subsequently, it was decided to create an independent test set to evaluate the generalization capabilities of the model and its final performance on completely unseen data. To this end, 10% of the images were extracted from the original training set and 10% from the original validation set, resulting in a new subdivision consisting of 447 training images, 90 validation images and 60 test images. The images for the new test set were extracted using random sampling.

2.2 Metrics for results evaluation

Four metrics were used to evaluate the performance of the vascular segmentation model: Dice Similarity Coefficient, Centerline Dice, Precision and Recall.

2.2.1 Dice Similarity Coefficient

Dice Similarity Coefficient (DSC) quantifies the degree of overlap between the predicted mask and the manual mask and varies between 0 (no match) and 1 (perfect match). It provides a global measure of segmentation quality, without distinguishing between over-segmentation and under-segmentation, summarizing both aspects in a single value. It is defined as:

$$DSC(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|}$$

where X represents the predicted mask and Y represents the manual mask.

2.2.2 Centerline Dice

Centerline Dice (clDice) was introduced to evaluate the topological correctness of tubular structures. It is defined as the harmonic mean between the Topological Precision (TP) and the Topological Sensitivity (TS):

$$TP(S_y, X) = \frac{|S_y \cap X|}{|S_y|}$$

$$TS(S_x, Y) = \frac{|S_x \cap Y|}{|S_x|}$$

$$clDice(Y,X) = \frac{2\, TP(S_y, X) \times TS(S_x, Y)}{TP(S_y, X) + TS(S_x, Y)}$$

where TP measures how much of the predicted skeleton is contained within the ground truth mask and TS quantifies how much of the ground truth skeleton is recovered by the predicted mask.

2.2.3 Precision and Recall

Precision and Recall were included because they allow to distinguish between over-segmentation and under-segmentation.

A low Recall indicates under-segmentation: the model fails to recover all the pixels belonging to vessels, due to a high number in false negatives (FN), especially losing the thinner or low-contrast structures.

$$Recall = \frac{TP}{TP + FN}$$

A low Precision indicates over-segmentation: the model produces a high number of false positives (FP), generating vessels which are not present in the ground truth or excessively expanding the real ones.

$$Precision = \frac{TP}{TP + FP}$$

2.3 Baseline definition

In order to segment vessels in retinal fundus images, a U-Net architecture was adopted, as it ensures effectiveness and strong performance in medical images segmentation. This model features a U-shaped structure composed of three main components: an encoder that extracts relevant features while reducing spatial resolution; a decoder that reconstructs the segmentation by restoring the original image resolution; skip connections that symmetrically link the encoder and decoder levels, which allows the recovery of spatial details lost during compression [4].

2.3.1 Hyperparameters of the U-Net

A key aspect of the project was the decision to input non-resized images (1024×1024 pixels) into the network. This choice was made to preserve fine details even of smallest vessels, which would be partially lost if the images were downscaled. However, this led to a significant increase in GPU memory consumption and so it was necessary to decrease the batch size to 2. The network depth was set to 4, representing a good balance between training stability and generalization. The number of initial filters was set to 24 to avoid memory saturation, after a preliminary test with 32 filters which showed that memory usage was nearly full. The model was trained for 20 epochs and the learning rate was set to 0.0006 to stabilize weight updates, which tend to become noisier when using a small batch size.

A DiceLoss was adopted as loss function because it directly optimizes DSC, ensuring consistency between training and validation objectives.

2.3.2 RGB and Grayscale

Initially, unprocessed RGB images were used. Subsequently, grayscale images were used as input to the U-Net while maintaining the same hyperparameters set before. This decision was motivated by several considerations. First, vessels are primarily identified by their shape and their contrast with the background, making pixels' intensity values sufficient for segmentation. In addition, grayscale images reduce computational complexity and eliminate redundant color information. Finally, this strategy is adopted in numerous studies in literature [5].

2.3.3 Green channel extraction

Moreover, in the field of medical image processing, the extraction of green channel is a standard practice for retinal fundus images segmentation, as retinal components appear more intense in this channel. Furthermore, it has the highest local contrast as compared to the red and blue channels, providing better visual separation between vessels and background [6]. The red channel is often oversaturated, while the blue channel has lower contrast and higher noise, making vessels boundaries harder to distinguish from the background.

The green channel extraction was evaluated using the same baseline architecture. This process involved the extraction of the green component from the RGB images and the consequent conversion to gray scale. As later discussed in the Results section, the performance metrics for the green channel turned out to be significantly lower than the gray scale ones. To understand the reasons behind these results, the green channel was visually compared to the gray scale. While the results for *Figure 2* align closely with the predictions, despite a higher-than-expected noise level in the green component, *Figure 3* demonstrates that underexposure leads to a significant loss of structural details. After doing a visual check of the dataset, the reason was attributed to the underexposure of a significant part of the images in the dataset.

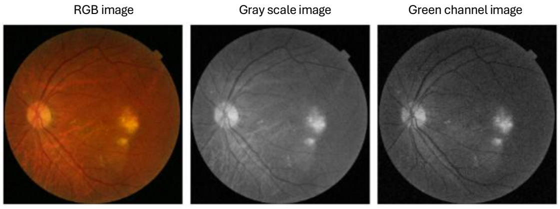

Figure 2: Effect of gray scale conversion and green channel extraction on a sample image ('004430.png')

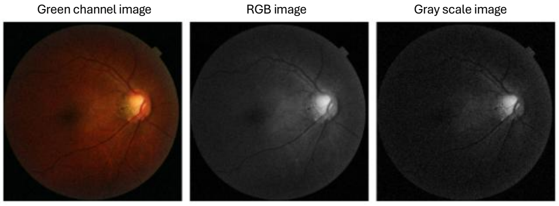

Figure 3: Effect of gray scale conversion and green channel extraction on a sample image ('003140.png')

2.3.4 Red and green channels fusion

To better understand which role each channel plays in the gray scale conversion, an individual analysis of each channel was conducted. As evidenced in *Figure 4* and *Figure 5*, the blue channel mainly captures high frequency noise and contributes to background non-uniformities due to increased light intensity at the borders. The green channel provides important structural information and high local contrast, but it is still affected by noise. Contrary to initial assumptions, the red channel does not suffer from overexposure and instead preserves significant structural information with an acceptable contrast and low noise. Consequently, exclusion of the red component led to bad performances, whereas the blue channel was considered useless for segmentation purposes. While standard RGB-to-grayscale conversion follows the weighted luma formula:

$$Y = 0.299R + 0.587G + 0.114B$$

a hybrid approach was adopted by combining the red and green channels, maintaining a ratio proportional to their original distribution*:*

$$Y = 0.337R + 0.663G$$

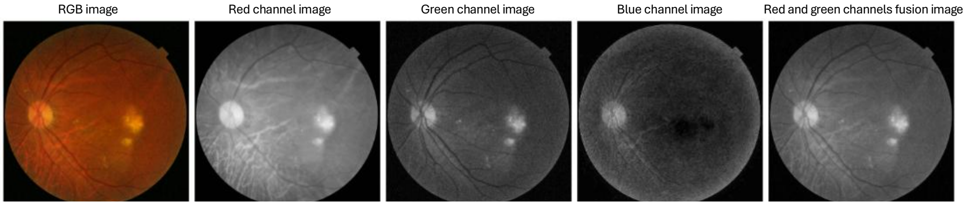

Figure 4: Extraction of the three channels from the RGB image and red and green channels combination on a sample image ('004430.png')

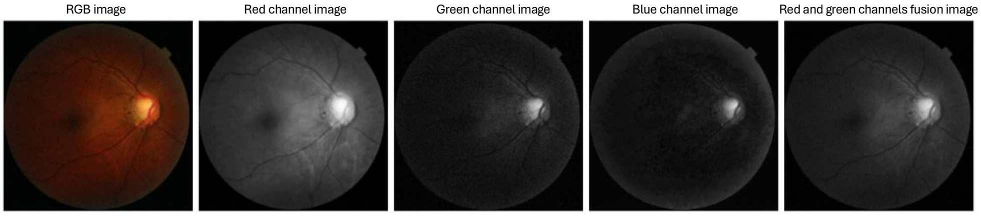

Figure 5: Extraction of the three channels from the RGB image and red and green channels combination on a sample image ('003140.png')

2.4 Preprocessing

Starting from the channel extraction described in the previous section, the following workflow addressed the noise and intrinsic variability of retinal fundus images. Four main stages were evaluated: denoising, light intensity adjustment, contrast enhancement and data augmentation techniques (applied only to the training set to improve generalization). Each stage was first independently evaluated on the baseline model and subsequently only the steps that appeared as most suitable for the segmentation task were combined to form the final preprocessing pipeline.

2.4.1 Denoising

Images derived from the channel extraction step exhibit a significant amount of noise. This variability is primarily attributable to electronic and thermal sensor noise, which is typically modeled as Additive White Gaussian Noise (AWGN) [7]. The presence of such noise poses a challenge for segmentation, as it can generate FPs in background regions or interrupt the continuity of thin vessels.

To mitigate this issue, different filtering techniques were compared, starting from an Averaging filter (kernel size of 3×3) which turned out to excessively blur the vessel edges. Non-linear approaches were also tested, such as a Bilateral filter, which turned out to significantly increase preprocessing time without yielding any significant improvement in the segmentation metrics and a Non-Local Means (NLM) filter, which is typically applied to preserve details, but instead led to a loss of thin vessels.

Finally, a Gaussian Filter with a kernel size of 3×3 and a $\sigma = 1$ emerged as the most effective solution.

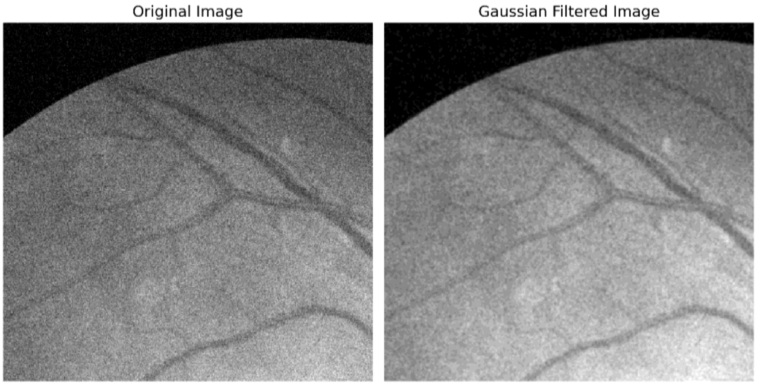

Figure 6: Effect of the Gaussian Filtering on the detail of a sample image ('141251.png')

As shown in *Figure 6*, as it aligns with the statistical distribution of the sensor noise, the Gaussian filter effectively suppresses the granular background texture while also preserving the structural integrity of the vascular tree.

2.4.2 Intensity Adjustment

Despite the initial channel extraction step, where the red and the green channels were combined to mitigate under-exposure as discussed before, the majority of the images in the dataset still exhibited an unbalanced intensity distribution, skewed toward dark tones. To assess this issue and ensure that vascular details were clearly distinguishable from the background, a Gamma Correction was introduced.

The transformation applied by the Gamma Correction is defined by the power-law relationship:

$$I_{out} = I_{in}^{\gamma}$$

Since the main problem was under-exposure, applying an exponent lower than one effectively "stretched" the mid and dark tones upward. However, a more aggressive intervention (e.g., $\gamma < 0.6$) would over-amplify amplified background noise and produce excessively bright and homogeneous images, so a conservative value of $\gamma = 0.9$ was chosen.

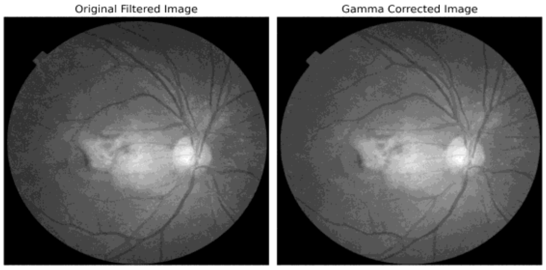

Figure 7: Effect of the Gamma Correction on a sample image ('141251.png')

As shown in *Figure 7*, this mild correction provided a light intensity increase while maintaining the structural integrity of the images. Since the applied value of Gamma is slightly lower than unity, the effect of the transformation can be mainly appreciated on darker areas of the background's image.

2.4.3 Contrast Enhancement

During the development of the preprocessing pipeline, Contrast Limited Adaptive Histogram Equalization (CLAHE) was tested to evaluate its impact on the algorithm's performances. The goal was to determine if an automated local contrast enhancement could further assist the network in identifying the vascular tree.
However, none of the tested combinations led to an increase in the segmentation metrics. Instead, results indicated that the inclusion of CLAHE led to an overall degradation of the model's performance. Since the previous steps (red-green channel fusion and Gamma Correction) already provided sufficient image clarity, the addition of CLAHE proved to be counterproductive and, for this reason, was excluded from the final preprocessing pipeline.

2.4.4 Data Augmentation

To improve the generalization capability of the network and to prevent overfitting, the size of the training set was tripled using data augmentation techniques. Specifically, geometric transformations were chosen as they preserve clinical characteristics of the image while introducing spatial variability [8].
The augmentation stage consists of four discrete transformations applied sequentially:

- a horizontal flip, which mirrors the image along the vertical axis;
- a vertical flip, that mirrors the image along the horizontal axis;
- a random 90° rotation, which randomly rotates the image by $90°$, $180°$ or $270°$;
- a transposition, that swaps the spatial axes (equivalent to a reflection across the main diagonal).

To strictly minimize the generation of duplicates (i.e., augmented images identical to the original) each transformation was applied with an independent probability of $P = 0.6$. This value of P limits the probability of two copies of the same image to be identical (not transformed):

$$\mathbb{E}\,[2\ identical\ copies] = \binom{3}{2} \times ((1-P)^4)^2\ \times P\ \times \#images = \ 3\ \times (0.4^4)^2\ \times (1-0.4^4)\ \times 447\ \ \approx 0.856\ images$$

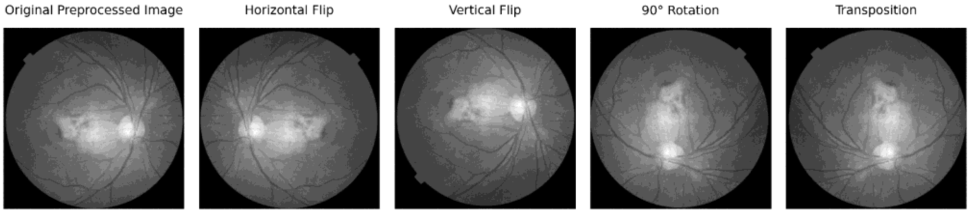

Figure 8: Visual representation of the possible four geometric transformations (horizontal flip, vertical flip, 90° rotation, transposition) applied on a sample image ('141251.png')

Figure 8 illustrates the effect of each transformation applied individually to a sample image for clarity. As explained before, these operations are not mutually exclusive: since each transformation is applied independently depending on its own probability P, a single image may undergo a sequence of multiple modifications (e.g., a horizontal flip followed by a 90° rotation).

2.5 Tuning

As mentioned in the previous sections, the initial learning curves showed significant fluctuations across epochs. This instability was attributed to the chosen value of the batch size, which was previously kept low due to the high memory demands of processing high-resolution images.
Batch size is critical for Batch Normalization (BN) layers, as they rely on batch-wise mean and variance to normalize activations. Small batches produce inaccurate statistics and stochastic noise, which directly destabilize the training process [9]. Increasing the batch size to 7 provides a more reliable statistical basis for the BN layers, allowing for a more stable training process, while also respecting the memory constraints.
To further stabilize the training without increasing memory usage, a 2-step gradient accumulation was introduced. Instead of updating the network's weights after every single batch, the gradients are calculated for two consecutive batches and accumulated before performing the backpropagation step. This reduction in stochastic noise resulted in a slower loss descent per epoch; consequently, the number of training epochs was increased from 20 to 30 to ensure full convergence.

Finally, to further manage memory constraints, the number of filters in the first layer was reduced from 24 to 16. This adjustment optimized memory usage during training, while still allowing the network to capture the essential features of the retinal vasculature.

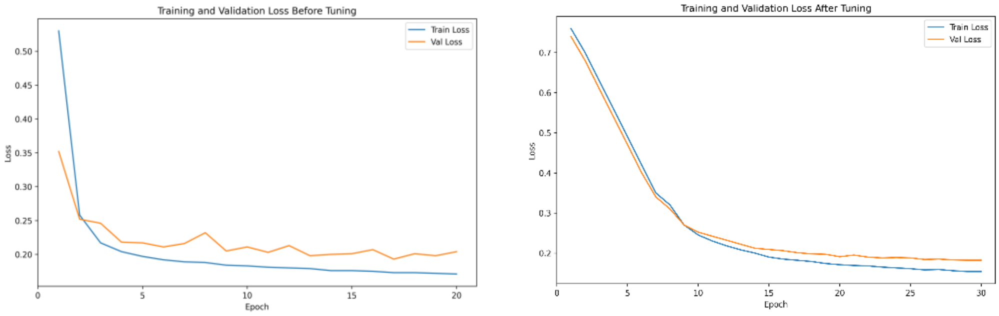

Figure 9: Effect of the tuning on the learning curves

As shown in *Figure 9*, these refinements led to smoother learning curves and a more robust convergence towards the global minimum of the loss function.

2.6 Post-processing

To refine the segmentation results, a post-processing pipeline was developed. Initially, a morphological closing operator was tested using a circular structuring element aiming to bridge narrow gaps. However, this approach proved unsuitable for retinal vessel refinement since a circular structural element lacks of directionality. In fact, it cannot distinguish between fragmented segments of the same vessel and adjacent parallel vessels *(Figure 10)*.

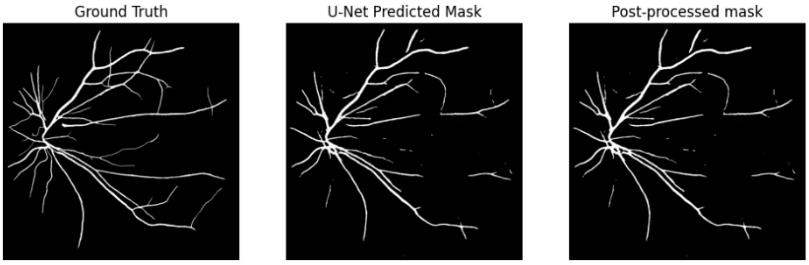

Figure 10: Effect of morphological closing on a sample U-Net mask ('144471.png')

To achieve not only better metrics performance, but also topologically coherent masks that are clinically readable and free from anatomical errors (such as the unintended merging of adjacent vessels), a different strategy was adopted. The binary segmentation masks were processed using morphological skeletonization to extract the vascular centerline and to identify vascular endpoints (*Figure 11)*, which serve as the basis for designing a more appropriate closing algorithm.

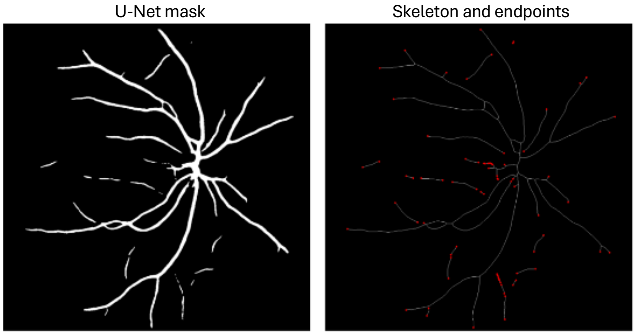

Figure 11: Visualization of the skeleton and the endpoints on a sample U-Net mask ('000675.png')

To address vessel fragmentation, the algorithm evaluates candidate links between endpoints by considering both the Euclidean distance and the angular alignment to ensure the continuity of the vessel. Furthermore, it reconnects isolated branches to longer vascular trunks based on a proximity threshold. Finally, the binary mask is reconstructed by integrating these newly formed bridges *(Figure 12)*. Unfortunately, the expected performance was not achieved. Along with an increase of the Recall, there was a significant decrease in the Precision. This indicates a disproportionate number of FPs compared to the gain in True Positives, suggesting that most of the bridges may have incorrectly connected non-existent vessels.

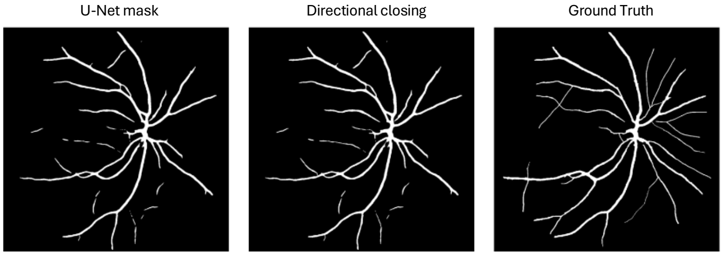

Figure 12: Effect of directional closing on a sample U-Net mask ('000675.png')

To prevent this issue, another strategy aimed at removing small components was proposed. The network might have misclassified noise as vessels and a high priority is placed on avoiding FPs. Since vessels are linear and narrow, a strategy based on the skeleton length was chosen rather than one based on area. Vascular structures with a skeleton length exceeding 40 pixels were retained *(Figure 13)*.
During testing, a higher precision was reached, but led to a lower recall. Under-segmentation was prioritized over the inclusion of non-vascular artifacts.

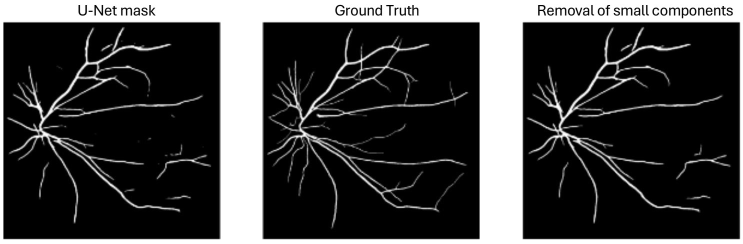

Figure 13: Effect of removing small components on a sample U-Net mask ('144471.png')

Several approaches were investigated by combining multiple post-processing techniques, such as integrating directional gap-filling with small components removal. However, experimental results indicated that these configurations did not yield any significant improvement. Considering these findings, the decision made was to maintain the simplest pipeline architecture by keeping only the small component removal.

3. Results

The following section presents the experimental results obtained during the development of the segmentation model. The performance of each stage is evaluated using the four metrics previously introduced. All values are reported as $mean \pm standard\ deviation$, rounded to four decimal places and calculated on the validation set at the best-performing epoch, to ensure an objective comparison of the different configurations. The results are organized into four main tables and the configurations highlighted in bold represent the specific modules or parameters selected for integration into the final processing pipeline.

	DSC	*clDice*	*Precision*	*Recall*
RGB	*0.8469 ± 0.1122*	*0.8457 ± 0.1210*	*0.8932 ± 0.1012*	*0.8122 ± 0.1355*
Grayscale	*0.7778 ± 0.1088*	*0.7622 ± 0.1205*	*0.8150± 0.0939*	*0.7335 ± 0.1347*
Green	*0.7528 ± 0.1075*	*0.7320 ± 0.1148*	*0.7870 ± 0.0980*	*0.7290 ± 0.1315*
Red + Green (lr = 0.0006)	*0.7729 ± 0.1051*	*0.7581 ± 0.1181*	*0.8025 ± 0.0919*	*0.7527 ± 0.1333*
Red + Green (lr = 0.0003)	***0.7760 ± 0.1087***	***0.7602 ± 0.1183***	***0.8133 ± 0.1042***	***0.7466 ± 0.1238***

Table 1: Baseline performances

As shown in *Table 1*, while the three-channel architecture (RGB) achieved higher metrics, the configuration based on the fusion of red and green channels was selected as the final baseline. Metrics values confirm that the inclusion of the blue channel provides a minimal and largely non-informative contribution compared to the other two channels. This behaviour is consistent with the previous discussion, as the blue channel is dominated by noise and non-uniformities in the background. The learning rate was lowered to 0.0003 to improve the stability of the learning curves.

	DSC	*clDice*	*Precision*	*Recall*
Averaging	0.7741 ± 0.1086	0.7583 ± 0.1199	0.8135 ± 0.1199	0.7577 ± 0.1302
NLM	0.7708 ± 0.1094	0.7523 ± 0.1258	0.8170 ± 0.0992	0.7411 ± 0.1316
Gaussian	**0.7768 ± 0.1084**	**0.7574 ± 0.1211**	**0.8200 ± 0.0988**	**0.7449 ± 0.1283**
Bilateral	0.7752 ± 0.1077	0.7594 ± 0.1199	0.8231 ± 0.0972	0.7452 ± 0.1283
Gamma	**0.7735 ± 0.1112**	**0.7538 ± 0.1287**	**0.8249 ± 0.1023**	**0.7357 ± 0.1373**
CLAHE	0.7656 ± 0.1097	0.7529 ± 0.1233	0.7880 ± 0.0984	0.7513 ± 0.1344
Data Aug.	**0.7888 ± 0.1093**	**0.7687 ± 0.1194**	**0.8308 ± 0.0924**	**0.7430 ± 0.1310**
Gaussian + Gamma + Data Aug.	**0.7894 ± 0.1069**	**0.7741 ± 0.1192**	**0.8273 ± 0.0922**	**0.7619 ± 0.1324**

Table 2: Pre-processing performances

Table 2 illustrates performances of the various preprocessing techniques evaluated against the baseline. As previously illustrated, among the denoising strategies, the Gaussian filter was selected as the optimal compromise between noise reduction and computational efficiency. Regarding contrast enhancement, CLAHE was excluded from the final pipeline as it failed to provide any measurable improvement over the results already achieved through Gamma Correction: instead, it led to a slight decrease in the segmentation metrics. The proposed solution for this stage integrates the Gaussian filter, Gamma Correction and Data Augmentation. As evidenced by the metrics in the table, Data Augmentation proved to be the most impactful component of the preprocessing workflow, leading to a significant increase in performance across all metrics, particularly in Precision.

	DSC	*clDice*	*Precision*	*Recall*
Tuning	*0.7915 ± 0.1076*	*0.7713 ± 0.1170*	*0.8319 ± 0.0876*	*0.7550 ± 0.1276*
Tuning + directional closing	*0.7914± 0.1045*	*0.7752± 0.1134*	*0.8217± 0.0821*	*0.7645± 0.1246*
Tuning + removal of small components	***0.7916 ± 0.1038***	***0.7775 ± 0.1148***	***0.8432 ± 0.0831***	***0.7519 ± 0.1239***

Table 3: Tuning and post-processing performances

Table 3 illustrates the results of the hyperparameters tuning and the post-processing techniques. As previously illustrated, tuning was applied not only to achieve better performance, but to provide better stability. During the post-processing definition, directional closing was discarded since it did not provide any improvement. The removal of small components did not apport any significant increase in DSC and clDice, but it improved Precision and decreased the Recall, as this trade-off was intended. Morphological closing was not included in this table since it generated erroneous vessel merges, which led to discard it.

	DSC	*clDice*	*Precision*	*Recall*
Final model on test set	**0.8108 ± 0.0738**	**0.8014 ± 0.0832**	**0.8546 ± 0.0380**	**0.7769 ± 0.1009**

Table 4: Test set performance

Table 4 illustrates the performance of the model applied to the test set, showing how the performance on the test set is slightly higher than the one observed on the validation set. This trend suggests good generalization ability of the model, but also that the metrics could be influenced by the different number of images in the two sets (90 in the validation and 60 in the test), as a smaller sample can produce less variable and sometimes more favorable estimates. Given that the test set was constructed via random sampling, it is also possible that it inadvertently included a higher proportion of images that are intrinsically easier to segment compared to those in the validation set.

4. Limitations and future developments

Although the model achieves good performance, it has several limitations that emerge during the development. Despite the fusion of the red and green channel, the use of high-resolution images, and the application of Gamma Correction, many images still lack an effective distribution of light intensity and are underexposed; consequently, the model finds it difficult to capture thin capillaries. Post-processing based on the removal of components with a skeleton smaller than 40 pixels reduced FPs, but did not demonstrate entirely effectiveness. It often eliminated small fragments of actual vessels, while larger FPs remained. This happened because of the impossibility to define a clear discriminant between small, disconnected vessels and broken vascular segments, with a resulting impact on the Recall. Furthermore, the chosen U-Net architecture, although it is a reliable standard, does not integrate advanced mechanism for fine detail recovery or for managing light variability.

These considerations suggest several possibilities for future developments: from an architectural point of view, the adoption of more recent U-Net variants, such as the Attention U-Net [10], could improve sensitivity towards thin vessels and increase the robustness in low-light conditions. A further improvement could come from the inclusion of light intensity normalization techniques. Post-processing with the removal of short components could be replaced or integrated with more selective strategies, able to better distinguish between noise and real capillaries, thus preserving a greater amount of vascular structure. Finally, using additional or multi-source datasets could increase the variability in acquisition conditions and improve the model's ability to generalize to images coming from different devices and lighting conditions.

References

[1] Ayoub, G. Disease Diagnosis Using Retinal Vasculature: Insights from Flammer Syndrome and AI. Brain Sci. 2025, 15, 919.

[2] Chen N, Zhu Z, Yang W and Wang Q (2024), Progress in clinical research and applications of retinal vessel quanti s fication technology based on fundus imaging. Front. Bioeng. Biotechnol.

[3] Md. Mohaimenul Islam, Tahmina Nasrin Poly, Bruno AndreasWalther, Hsuan Chia Yang and Yu-Chuan (Jack) Li. Artificial Intelligence in Ophthalmology: A Meta-Analysis of Deep Learning Models for Retinal Vessels Segmentation.

[4] Ronneberger, O., Fischer, P., Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol. 9351. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28

[5] Liskowski, P., & Krawiec, K. (2016). Segmenting retinal blood vessels with deep neural networks. IEEE Transactions on Medical Imaging, 35(11), 2369–2380.

[6] Abdulsahib, A.A., Mahmoud, M.A., Mohammed, M.A. et al. Comprehensive review of retinal blood vessel segmentation and classification techniques: intelligent solutions for green computing in medical images, current challenges, open issues, and knowledge gaps in fundus medical images. Netw Model Anal Health Inform Bioinforma 10, 20 (2021).

[7] N. Vannadil and P. Kokil, "Noise and performance analysis on fundus images with CNN and transformer models," 2023 IEEE 7th Conference on Information and Communication Technology (CICT), Jabalpur, India, 2023.

[8] Garcea F., Serra A., Lamberti F., Morra L., Data augmentation for medical imaging: A systematic literature review, Computers in Biology and Medicine, Volume 152, 2023, 106391

[9] Yong, H., Huang, J., Meng, D., Hua, X., Zhang, L. (2020). Momentum Batch Normalization for Deep Learning with Small Batch Size. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science(), vol 12357. Springer, Cham.

[10] Tsai, CM., Qiu, SH., Chang, CW. (2026). Retinal Vessel Segmentation Using an Attention-Enhanced U-Net Architecture. In: Kim, BG., Sekiya, H., Lee, D. (eds) Multimedia Information Technology and Applications. MITA 2025. Communications in Computer and Information Science, vol 2675. Springer, Singapore.