

# Projet 1

## Segmentation automatique du réseau vasculaire dans des images de rétine

AILLAUD Arnaud

21 mars 2023

### Table des matières

<b>1 Résultats</b>	<b>2</b>
<b>2 Questions et pistes de réflexion</b>	<b>3</b>
<b>3 Comportement et limites de l'algorithme</b>	<b>4</b>
3.1 Démarche . . . . .	4
3.1.1 Pre-processing . . . . .	4
3.1.2 Segmentation grossière . . . . .	5
3.1.3 Affinage via convolutions . . . . .	5
3.1.4 Post-processing . . . . .	5
3.2 Limites . . . . .	6

# 1 Résultats

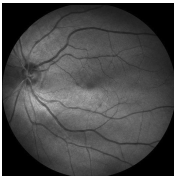
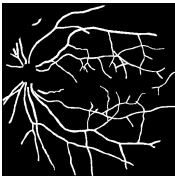

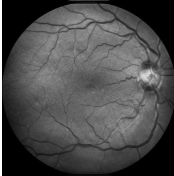
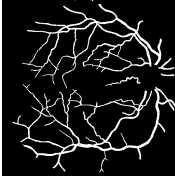

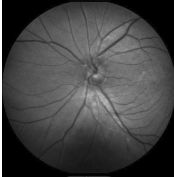
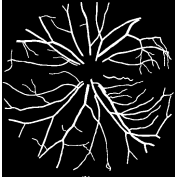
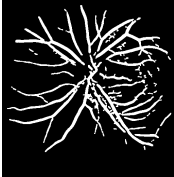
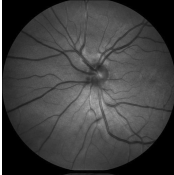
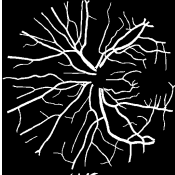
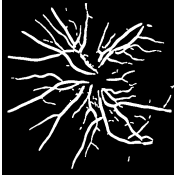
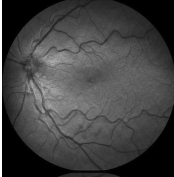
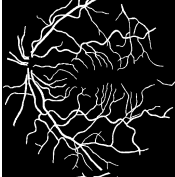
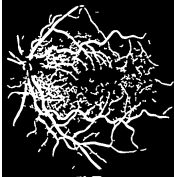
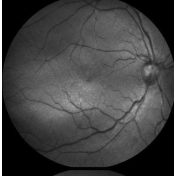
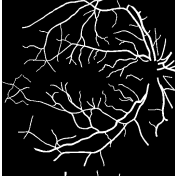
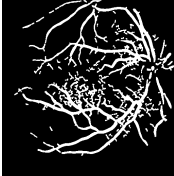
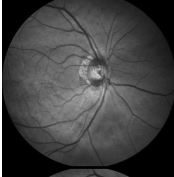
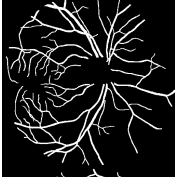
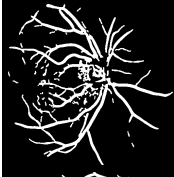
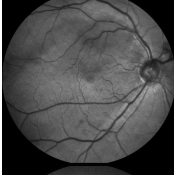
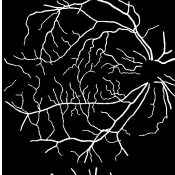

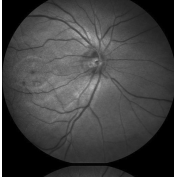
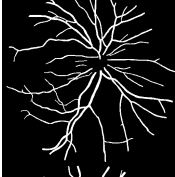
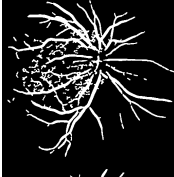
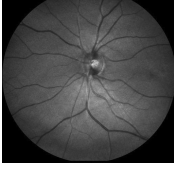
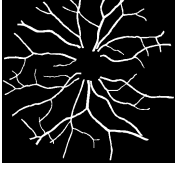
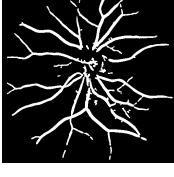
Nom	Image originale	Vérité terrain	Image segmentée	Precision	Rappel	F1-Score
Star01				0.766	0.765	0.765
Star02				0.745	0.846	0.792
Star03				0.854	0.687	0.761
Star08				0.941	0.650	0.769
Star21				0.608	0.741	0.668
Star26				0.685	0.728	0.706
Star28				0.807	0.642	0.715
Star32				0.821	0.782	0.801
Star37				0.711	0.762	0.736
Star48				0.871	0.763	0.814

TABLE 1 – Résultats de segmentation pour les 10 images du jeu de données

Les résultats de l'algorithme de segmentation sont présentés dans le tableau ci-dessus. Ils peuvent être résumés par :

1. Un score de précision moyen de **0.78**
2. Un score de rappel moyen de **0.74**
3. Un F1-Score moyen de **0.75**

## 2 Questions et pistes de réflexion

*Expliquez la fonction d'évaluation fournie dans le script Python. Pourquoi utilise-t-on deux métriques (Précision et Rappel) ? Quel rôle joue la squelettisation dans cette fonction d'évaluation ?*

La fonction d'évaluation fournie dans le script Python compare une image segmentée via l'algorithme à une image annotée par un expert. Via cette comparaison sont calculées 3 valeurs :

1. Le nombre de faux positifs FP : pixels affichés en blanc sur l'image segmentée mais n'appartenant pas à un vaisseau sanguin. FP est calculé via l'opération "ET logique" entre l'image segmentée et l'inverse de l'image de référence
2. Le nombre de faux négatifs FN : pixels affichés en noir sur l'image segmentée, et appartenant en réalité à un vaisseau sanguin. FN est calculé via l'opération "ET logique" entre l'image de référence et l'inverse de l'image segmentée
3. Le nombre de vrais positifs TP : pixels appartenant à un vaisseau sanguin correctement affichés en blanc dans l'image segmentée. TP est calculé via l'opération "ET logique" entre l'image segmentée et l'image de référence

Ces valeurs servent à calculer deux métriques :

$$\text{Précision} = \frac{TP}{TP + FP} \text{ et } \text{Rappel} = \frac{TP}{TP + FN}$$

Maximiser une seule de ces métriques ne garantit pas d'avoir un algorithme pertinent. En effet, la précision est maximale tant qu'aucune erreur n'est commise. Un algorithme qui ne classerait qu'un seul point sur l'image correctement aurait donc une précision maximale. De même, le rappel est maximal s'il n'y a pas de faux négatifs, ce qui est le cas si aucune prédiction négative n'est faite : une image entièrement blanche garantirait un rappel de 100%.

C'est pourquoi ces 2 métriques sont utilisées conjointement. Elles peuvent être combinées pour obtenir le F1-score, ce qui permet de ne travailler qu'avec une seule métrique (plus simple pour optimiser et comparer des résultats).

Une autre particularité de cette fonction d'évaluation est l'utilisation de la squelettisation sur l'image segmentée dans le calcul de FP et TP, et sur l'image de référence dans le calcul de FN. Comme illustré ci-dessous, la squelettisation permet d'affiner les traits. Ainsi, la comparaison de l'image segmentée squelettisée avec la vraie image de référence améliore FP (moins de pixels blancs sur l'image après squelettisation), TP (car comparaison avec l'inverse de l'image de référence) et FN (également car comparaison avec l'inverse de l'image segmentée). La squelettisation permet donc d'améliorer légèrement les scores obtenus, en supprimant du bruit sur les images à comparer / permettant de générer des images avec des traits plus épais et donc de boucher les trous des vaisseaux sanguins sans impact.



FIGURE 1 – Image segmentée



FIGURE 2 – Image segmentée squelettisée

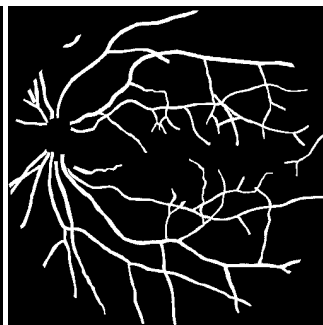


FIGURE 3 – Image de vérité terrain

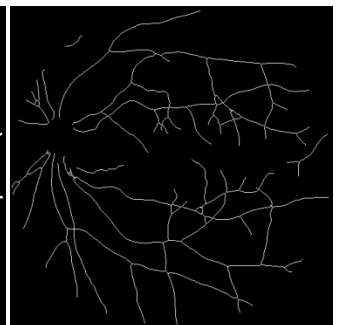


FIGURE 4 – Vérité terrain squelettisée

### 3 Comportement et limites de l'algorithme

#### 3.1 Démarche

L'algorithme se décompose en plusieurs étapes :

1. Pre-processing
2. Segmentation grossière
3. Affinage via convolutions
4. Post-processing

La première image du jeu de données est utilisée pour illustrer la démarche.

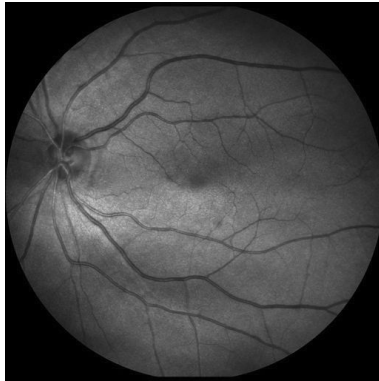


FIGURE 5 – Image initiale

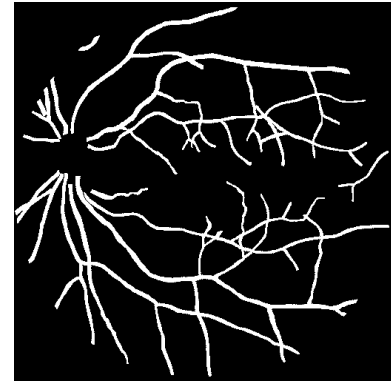


FIGURE 6 – Vérité terrain

##### 3.1.1 Pre-processing

La première étape de pre-processing consiste à supprimer une partie du bruit de l'image initiale. Un filtre médiane a été appliqué pour cela, car il donnait empiriquement de meilleurs résultats qu'un filtre gaussien.

L'image générée semble très similaire à l'image initiale, mais cette étape de débruitage a bien un impact non négligeable sur le nombre de tâches blanches visibles sur les images suivantes par la suite.

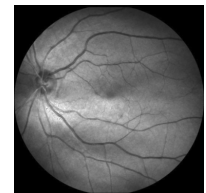


FIGURE 7 – Filtrage médiane

Le fond étant clair de manière assez uniforme, et les vaisseaux sanguins foncés, une transformation top-hat permet de réaliser un premier niveau de segmentation. Un disque de rayon 4 a été choisi comme élément structurant. L'augmentation du rayon permet de faire ressortir plus facilement les vaisseaux sanguins ... mais également l'iris, et d'autres tâches blanches parasites (bord de l'oeil). Une valeur entre 4 et 8 semble un bon compromis. Une étape de remise à l'échelle du niveau de gris via l'algorithme CLAHE permet par la suite de mieux faire ressortir les vaisseaux. Cet algorithme améliore le contraste même dans les régions très sombres ou éclairées de l'image. Une taille de kernel de 32 donne de meilleurs résultats que la valeur par défaut (64 dans notre cas pour des images de taille 512 x 512 pixels).

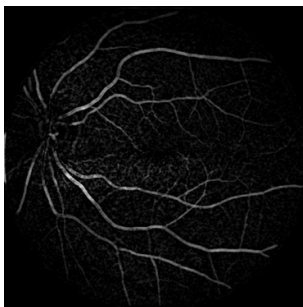


FIGURE 8 – Application du black top-hat

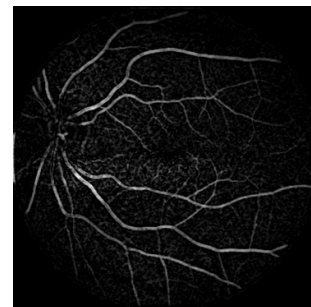


FIGURE 9 – Augmentation du contraste avec CLAHE

### 3.1.2 Segmentation grossière

L'augmentation du contraste rend certes les vaisseaux plus visibles, mais augmente également le bruit de l'arrière-plan. L'application d'un filtre de Meijering, avec un range de 1 à 5, permet de faire diminuer le bruit de fond et de bien délimiter les vaisseaux.

L'application d'un seuil d'Otsu est ensuite utilisée pour binariser et finir de nettoyer l'image.

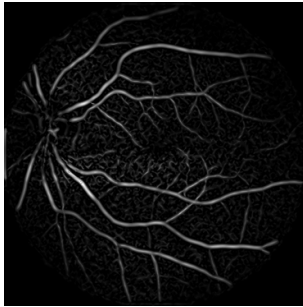


FIGURE 10 – Application du filtre de Meijering

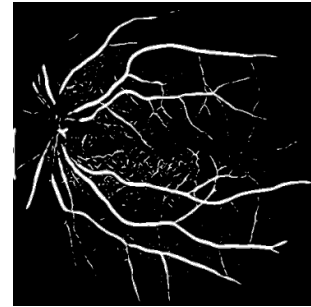


FIGURE 11 – Seuillage via méthode d'Otsu

A partir de cette étape, les vaisseaux épais sont bien segmentés, et le bruit de fond semble relativement aisé à supprimer via une ouverture. Le challenge se porte donc plutôt sur la segmentation des vaisseaux fins, qui seront perdus si on réalise effectivement une ouverture pour supprimer le bruit.

### 3.1.3 Affinage via convolutions

Pour segmenter les vaisseaux sanguins plus fins, j'ai réalisé 4 filtres linéaires,  $K_1$  à  $K_4$  représentés ci-dessous, pour récupérer les lignes horizontales, verticales et diagonales. Pour détecter les petits vaisseaux, un nombre minimal de lignes doit contenir des valeurs positives dans les filtres. J'ai testé plusieurs tailles de filtre : des kernels de taille 3x3 donnent des résultats convenables.

$$K_1 = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \quad K_2 = \begin{pmatrix} 0 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & 0 \end{pmatrix} \quad K_3 = \begin{pmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{pmatrix} \quad K_4 = \begin{pmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{pmatrix}$$

Travailler avec la somme des 4 images obtenues permet de recréer une partie des petits vaisseaux sanguins.

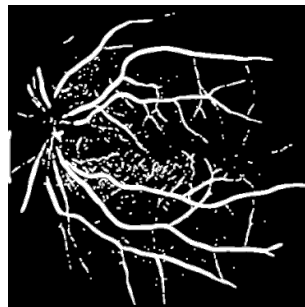


FIGURE 12 – Résultat après combinaison des 4 convolutions

### 3.1.4 Post-processing

La dernière étape consiste à finir de nettoyer l'image : suppression des artefacts, des bruits parasites, etc. Un masque rectangulaire est appliqué pour supprimer les artefacts aux bords de l'image.

Une fermeture est ensuite appliquée (élément structurant disque de rayon 1) pour tenter de boucher les trous au milieu des vaisseaux sanguins sans trop déformer l'image. Finalement, une ouverture est appliquée via `diameter_opening` de diamètre 9, pour tenter de supprimer les dernières traces de bruit présentes au milieu de l'image.



FIGURE 13 – Résultat final après post processing

L'ensemble de la démarche est repris dans le notebook contenant le code.

### 3.2 Limites

L'algorithme développé comporte tout de même plusieurs limites

1. La tentative de segmenter les vaisseaux fins engendre du bruit difficile à nettoyer sur la plupart des images. Il semble difficile de réussir à supprimer le bruit sans effacer les vaisseaux sanguins de petite taille

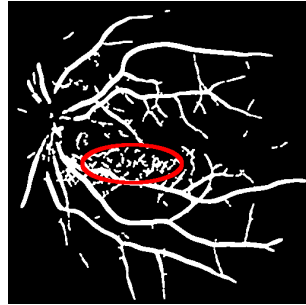


FIGURE 14 – Exemple de bruit résiduel

2. Des vaisseaux sanguins sont détectés dans le coeur de l'iris (espace qui reste toujours vide sur les images annotées). Une piste pour améliorer ce problème peut être l'utilisation d'opérateurs morphologiques pour trouver un cercle de la taille de l'iris

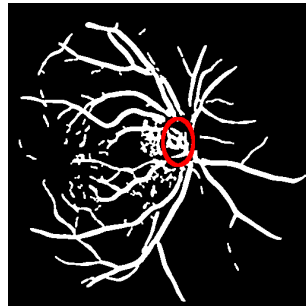


FIGURE 15 – Exemple d'iris présent sur l'image segmentée (Star28)

3. Malgré l'utilisation de CLAHE pour effectuer une amélioration du contraste local, les vaisseaux sanguins en bordure d'oeil restent moins bien détectés : ils sont sombres sur fond sombre. Peut-être que d'autres algorithmes de thresholding ou de filtrage permettraient d'augmenter le contraste en bordure d'oeil pour essayer de traiter ce problème



FIGURE 16 – Exemple de vaisseaux sanguins difficiles à contraster