

Rapport TP2, cours MI206

I. Traitement de l'image avant la segmentation

Parfois les vaisseaux dans l'image ne sont pas assez évidents. C'est mieux de traiter l'image pour que les vaisseaux soient plus faciles à détecter. Nous choisissons des méthodes ci-dessous :

- **Égaliser l'histogramme d'une image en niveaux de gris.** L'égalisation de l'histogramme consiste à étirer l'histogramme d'origine afin qu'il soit uniformément réparti dans toute la gamme de niveaux de gris. Elle peut améliorer ainsi le contraste de l'image. Voici un exemple avant et après l'égalisation de l'histogramme :



Figure 1 - Exemple avant et après l'égalisation de l'histogramme

La fonction correspondante est `cv2.createCLAHE(double clipLimit, Size tileGridSize)`. Cette fonction peut limiter le contraste. Après nous utilisons `clahe.apply(image)`.

- **Erosion de l'image.** Comme les vaisseaux sont plutôt noirs dans l'image, l'érosion peut faire les vaisseaux plus évidents. Tout d'abord, nous utilisons `erosion(img,selem=disk(2))` pour que les vaisseaux soient plus gros. Ensuite, nous utilisons `erosion(img,selem=rectangle(1,3))` et `erosion(img,selem=rectangle(3,1))` pour que les vaisseaux soient plus 'connectés'. En fait, c'est mieux de appliquer l'érosion par l'élément structurant de segment de toutes les directions car les vaisseaux peuvent être aux directions variées. Mais nous ne trouvons pas le code pour utiliser un segment d'une certaine direction comme l'élément structurant. Donc nous n'utilisons que les directions horizontale et verticale.

A partir de traitements ci-dessus, nous obtenons une image plus claire. Voici un exemple :

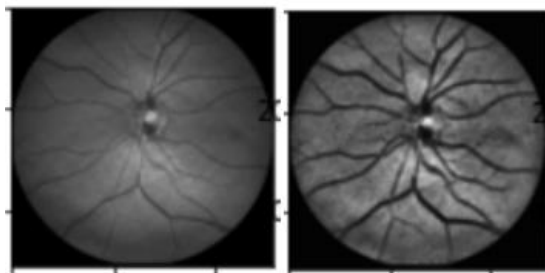


Figure 2 - Exemple avant et après les traitements

II. Segmentation

La segmentation est pour dessiner les vaisseaux dans l'image. Nous utilisons des méthodes ci-dessous à l'ordre :

- **Seuil**. Un seuil permet d'avoir une initialisation de la segmentation. Tous les pixels qui ont une valeur inférieure au seuil sont gardés. Après avoir testé sur des méthodes différentes, nous choisissons la fonction `filters.threshold_sauvola(img,int window_size, double k, int r)` qui a une meilleure performance. Après l'application du seuil, l'image est devenu comme :

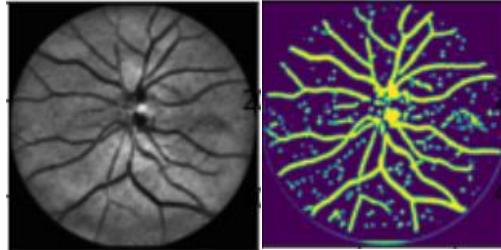


Figure 3 - Application du seuil

- **Erosion**. Comme nous avons grossi les vaisseaux dans la partie I, il faut diminuer leur épaisseur pour qu'ils reviennent à l'épaisseur originale. Ici nous utilisons encore l'érosion, car cette fois les vaisseaux sont blancs, et nous voulons diminuer leur épaisseur. La fonction utilisée est `erosion(img, disk(2))`.

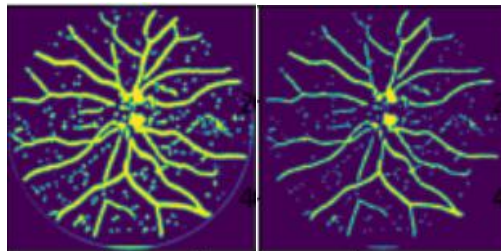


Figure 4 - Erosion pour revenir à l'épaisseur originale

Nous constatons que la partie centrale des vaisseaux est particulièrement grosse, et nous ne la voulons pas. Nous utilisons encore une fois l'érosion pour éliminer cette partie. Tout d'abord, nous érosons l'image `img1` pour avoir la partie centrale `img2`. La fonction correspondante est `erosion(img_out, disk(5))`. Ensuite, nous faisons une soustraction entre `img1` et `img2` par `img1^img2` (Elles sont de valeurs booléennes) .

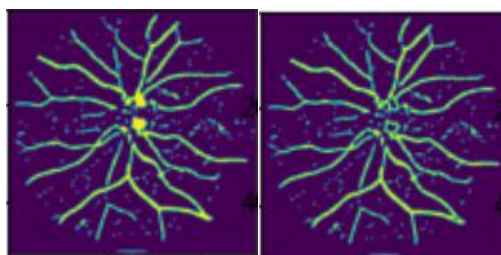


Figure 5 - Supprimer la partie centrale des vaisseaux

- **Supprimer de petits objets.** Dans la Figure 5, il y a beaucoup de petits objets (bruits) dont nous n'avons pas besoin. Donc nous utilisons la fonction **remove_small_objects(img_out, int min_size)** pour les supprimer. Voici le résultat :

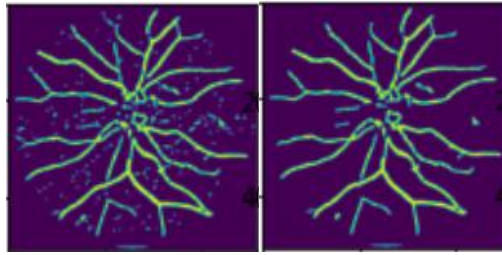


Figure 6 - Supprimer de petits objets

III. Résultats

Après la segmentation, nous pouvons évaluer des résultats obtenus. Ici nous utilisons 2 paramètres : la précision (accuracy) et le rappel (recall) . La précision nous dit combien de prédit pixels blanc sont correctes dans notre segmentation, alors que le rappel nous dit combien de pixels blanc dans la vraie segmentation sont inclus dans notre segmentation. Tous les deux paramètres sont importants et nous ne pouvons pas supprimer un entre eux. Il est possible que la précision soit très haut mais le résultat soit mauvais. Un cas extrême est qu'il y n'a qu'un pixel blanc dans notre segmentation et il est correct. Et il est possible que le rappel soit très haut mais le résultat soit mauvais. Par exemple, tous les pixels sont blanc dans notre segmentation.

Nous utilisons les squelettes dans l'évaluation pour donner un peu de tolérance sur le résultat. Souvent l'épaisseur des vaisseaux dans le résultat ne sont pas homogène, et il est différent de celui de la vraie segmentation, mais la forme est le même. Dans ce cas, nous pouvons encore dire que le résultat est bon.

Les précisions et les rappels des 10 images différentes sont dans le tableau ci-dessous. Nous ajoutons aussi les images de segmentation dans l'annexe.

	Précision (Accuracy)	Rappel (Recall)
star01_OSC	95.6	90.9
star02_OSC	94.5	94.1
star03_OSN	97.9	88.6
star08_OSN	97.7	89.7
star21_OSC	95.8	85.0
star26_ODC	95.1	87.1
star28_ODN	94.8	87.8
star32_ODC	92.6	86.8
star37_ODN	96.5	91.4
star48_OSN	97.0	92.6

Tableau 1 - Résultat des 10 images différentes

IV. Limites

Bien que le résultat obtenu ne soit pas mauvais, nous n'avons pas pu avoir un meilleur résultat. Dans ce TP, nous n'avons pas trouvé une méthode pour connecter des segments séparés des vaisseaux. De plus, le rappel est plus petit que la précision, c'est parce qu'une partie de vaisseaux dans l'image est si fine et de couleur claire que la segmentation ne peut pas les détecter, et parce que une partie de segmentation est perdue lors de l'élimination de bruits. Par ailleurs, il y a des bruits que nous n'avons pas pu éliminer même si leur taille est petite. Si nous avons plus de temps, nous pouvons concentrer aux problèmes ci-dessus et avoir un meilleur résultat.

V. Annexe - les images de segmentation

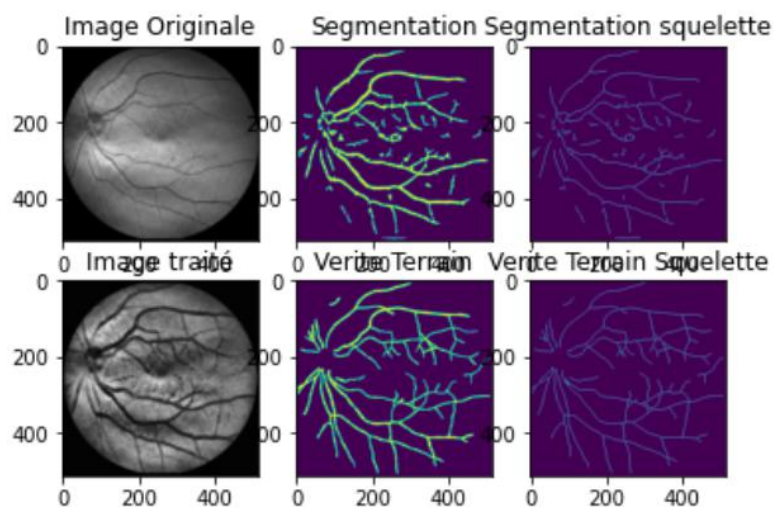


Figure 7 - star01_OSC (Précision : 95.6, Rappel : 90.9)

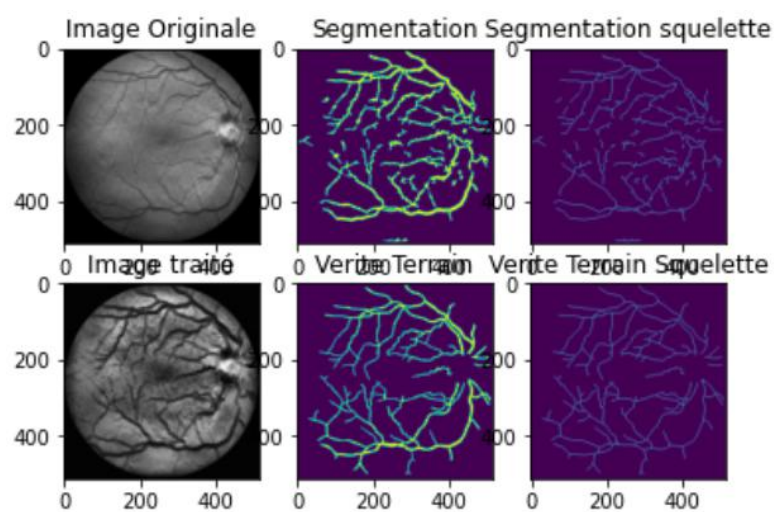


Figure 8 - star02_OSC (Précision : 94.5, Rappel : 94.1)

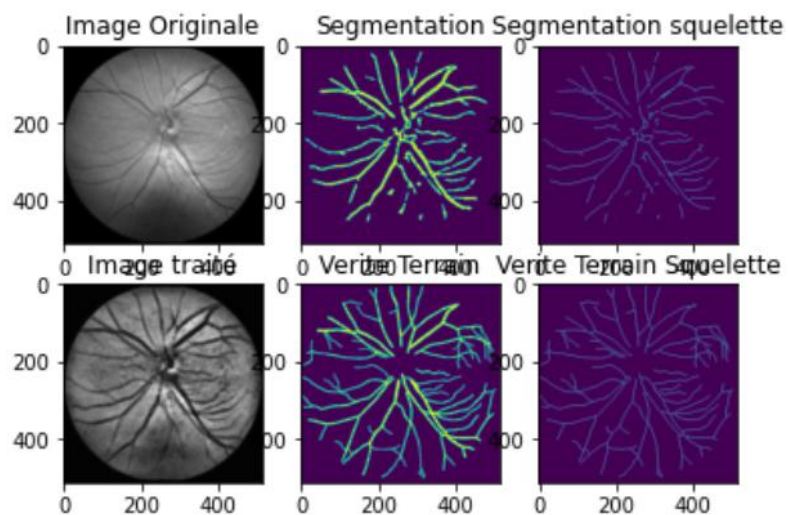


Figure 9 - star03_OSN (Précision : 97.9, Rappel : 88.6)

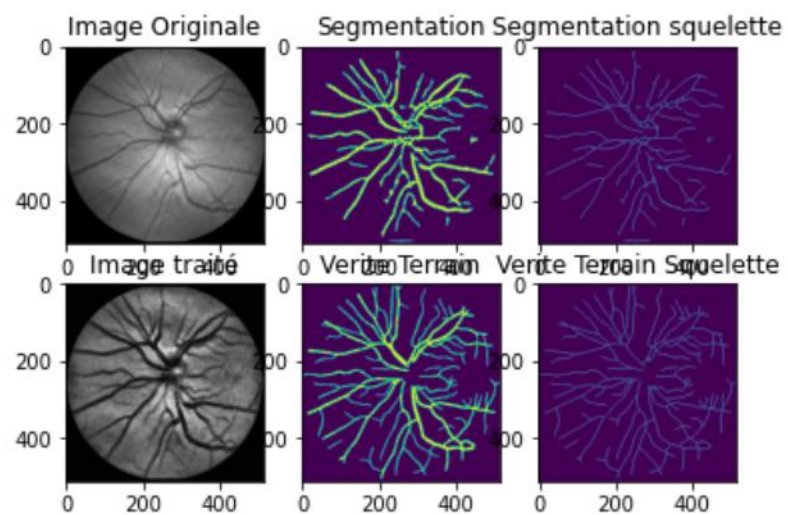


Figure 10 - star08_OSN (Précision : 97.7, Rappel : 89.7)

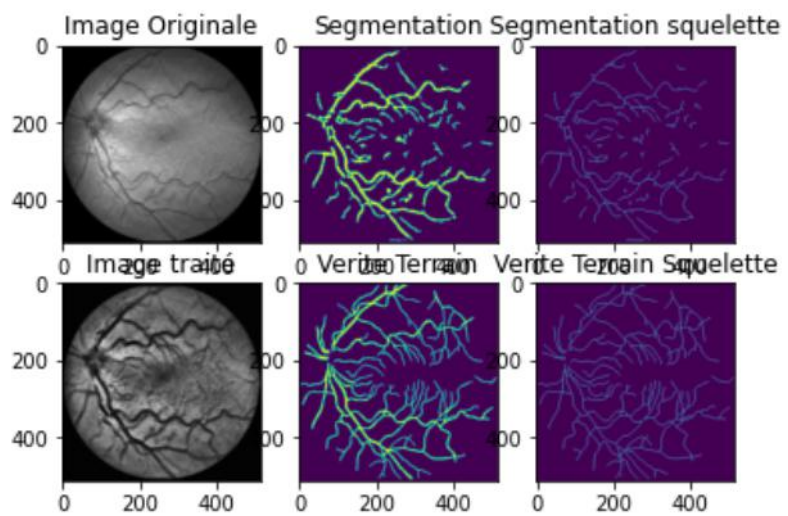


Figure 11 - star21_OSC (Précision : 95.8, Rappel : 85.0)

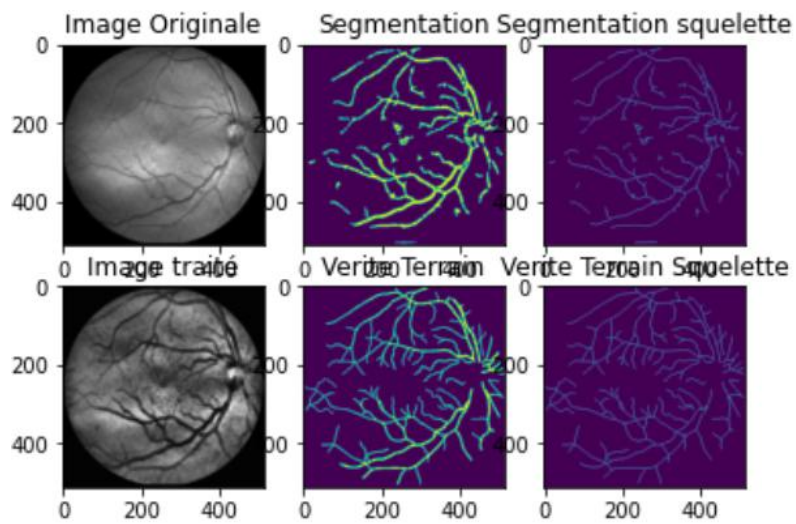


Figure 12 - star26_ODC (Précision : 95.1, Rappel : 87.1)

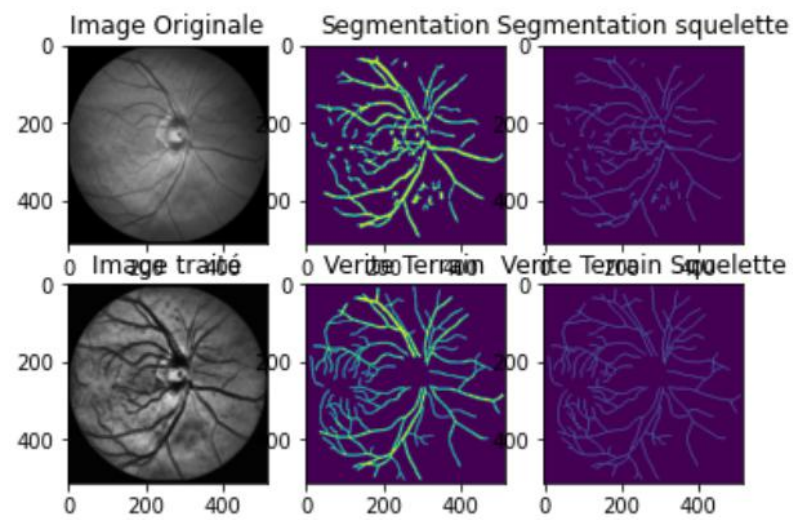


Figure 13 - star28_ODN (Précision : 94.8, Rappel : 87.8)

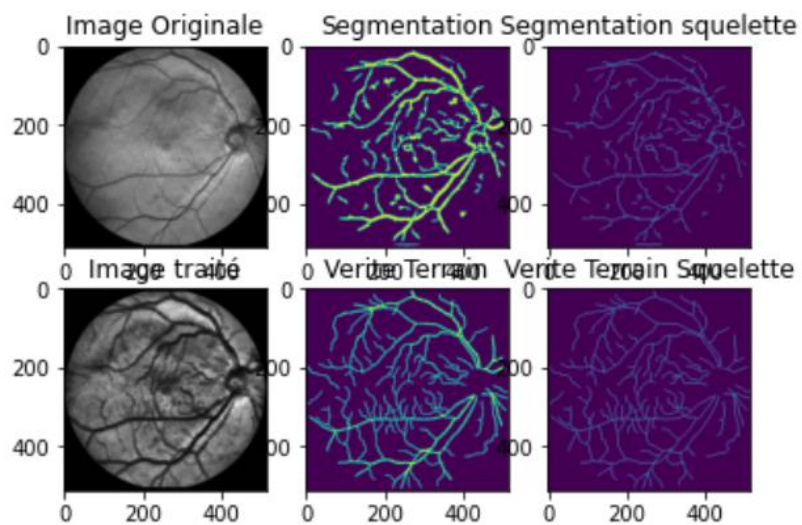


Figure 14 - star32_ODC (Précision : 92.6, Rappel : 86.8)

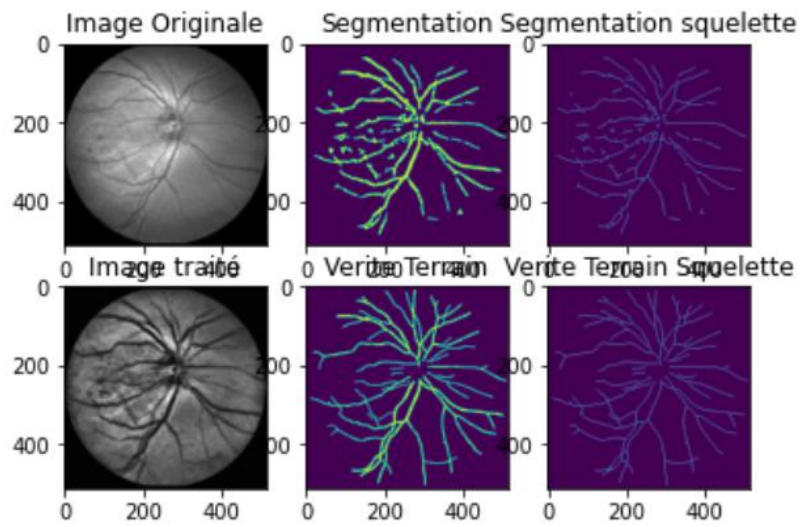


Figure 15 - star37_ODN (Précision : 96.5, Rappel : 91.4)

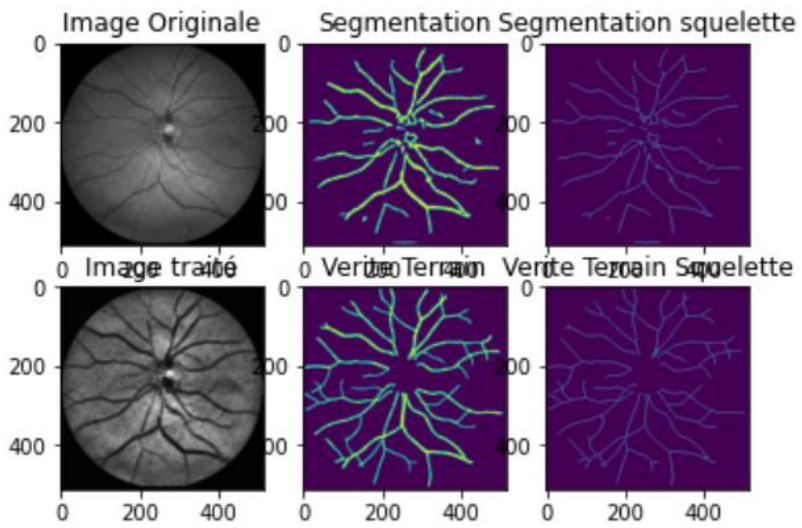


Figure 16 - star48_OSN (Précision : 97.0, Rappel : 92.6)