

Projet Vision par Ordinateur

BACRY Ilan - MOREAU Lucie - TOLEDANO Chloé

Table des matières

I.	Introduction.....	3
II.	Littérature scientifique	5
1.	Tomographie par cohérence optique (OCT)	5
2.	Photographie stéréoscopique de la papille optique.....	5
3.	Coloration histologique et immunohistochimie	5
4.	Apprentissage automatique et intelligence artificielle.....	5
III.	Pré-traitement de l'image	6
1.	Elimination des rectangles noirs.....	6
2.	Augmentation du contraste.....	6
a.	Manipulation à la main	6
b.	Egalisation d'histogramme	7
c.	Top hat et bottom hat	8
3.	Lissage	8
IV.	Méthodes de segmentation	9
1.	Seuillage	9
a.	Seuil fixé par l'utilisateur.....	9
b.	Méthode d'Otsu	10
c.	Seuil gaussien.....	10
2.	Seuillage par hystérésis	11
3.	Détection des contours	11
a.	Noyaux de Sobel.....	12
b.	Filtre de Canny	12
V.	Amélioration de l'image	13
1.	Ouverture et Fermetures.....	13
2.	Filtres alternés séquentiels.....	14
3.	Comblir les trous	14
VI.	Expérimentation.....	15
1.	Méthode de seuillage	16
2.	Méthode d'ouverture et fermeture.....	18
3.	Méthode des Top Hat	21
IV.	Comparaison.....	23
VII.	Intelligence Artificielle.....	24
1.	Convolutional Neural Network.....	24
2.	U-Net	25
VIII.	Conclusion	27

I. Introduction

Le glaucome est une maladie chronique de l'œil, causée par des lésions du nerf optique qui perd progressivement ses fibres. Elle apparaît souvent avec l'augmentation de la pression intraoculaire, entraînant une perte progressive du champ visuel, commençant par les bords. Sans traitement, elle peut mener à une déficience visuelle. Le glaucome survient généralement après 40 ans, touchant 1 à 2 % des personnes au-delà de cet âge et 10 % des personnes au-delà de 70 ans.

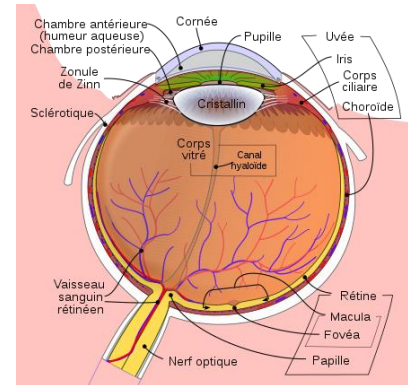


Figure 1: Anatomie d'un œil humain

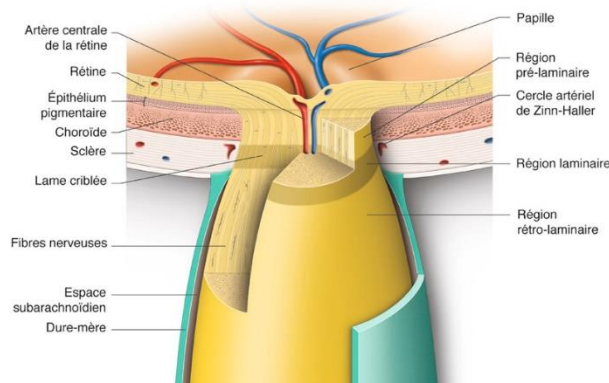


Figure 2: Schéma d'une coupe de la tête du nerf optique (papille) et du canal scléral

La lame criblée, située dans la tête du nerf optique, joue un rôle crucial en servant de passage aux fibres nerveuses qui connectent la rétine au cerveau. En raison de sa fonction essentielle et de sa complexité structurale, elle est susceptible d'être affectée par diverses pathologies.

L'application de l'Optique Adaptative (OA) dans l'imagerie de la lame criblée fournit une résolution extrêmement élevée, permettant ainsi de révéler les détails subtils de cette structure. Toutefois, la segmentation précise des pores de la lame reste un grand défi.

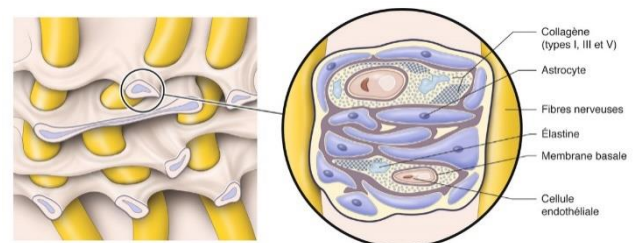


Figure 3: Lame criblée vue de près, un tamis à plusieurs étages

L'objectif de ce projet est de développer des algorithmes de traitement d'image afin de segmenter au mieux les pores de la lame criblée (LC). L'implémentation de l'automatisation demande plusieurs étapes, notamment de pré-traitement d'image avant toute opération de segmentation, puis possiblement l'utilisation de Machine Learning pour essayer d'obtenir les paramètres les plus adaptés pour parfaire l'algorithme.

Pour ce projet notre base de données est constituée de 10 images en niveaux de gris d'OA de lame criblée appartenant à des patients atteints de glaucome, et leurs 10 vérités terrain associées, c'est-à-dire des masques dessinés par un ophtalmologiste entourant les zones d'intérêt pour la détection du glaucome. Le but est d'automatiser la détection de ces zones d'intérêt pour accélérer et faciliter la détection de la maladie.

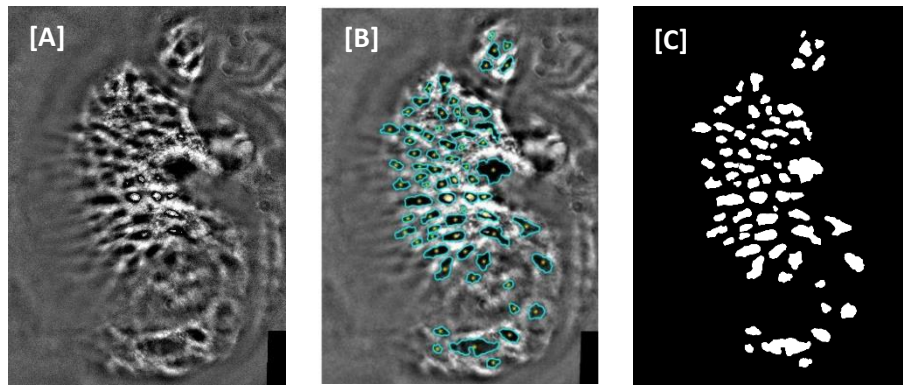


Figure 4 : **[A]** Image originale de la LC1 **[B]** Image avec détection des contours sur la LC1
[C] Vérité terrain des zones à détecter sur la LC1

Parmi les complexités de cet exercice, les images fournies n'ont pas les mêmes tons de gris, des histogrammes très différents, des cadrages variés, des formes aléatoires, et certaines possèdent même des rectangles noirs qui bloquent une partie du champ de vision.

Afin de faciliter la compréhension du rapport, l'abréviation « LC » sera utilisée pour désigner la lame criblée, les chiffres qui suivent correspondront aux numéros dans notre base de données. Par exemple, la LC1 fera référence à l'image de lame criblée nommée LC0001.

II. Littérature scientifique

Aujourd'hui, le traitement des images pour le diagnostic et l'étude du glaucome repose sur quelques méthodes avancées que nous allons décrire ci-dessous :

1. Tomographie par cohérence optique (OCT)

L'OCT est une méthode non invasive, couramment utilisée, qui permet de réaliser des coupes transversales des structures rétinienne et du nerf optique. Elle fournit des informations précises sur l'épaisseur des couches rétinienne et des fibres nerveuses, essentielles pour détecter le glaucome de manière précoce.

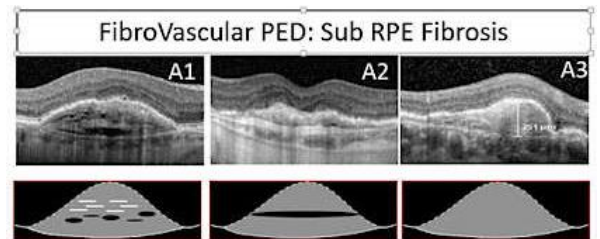


Figure 5: Différents morphotypes des lésions fibreuses en OCT haute résolution

2. Photographie stéréoscopique de la papille optique

Cette méthode permet une visualisation en trois dimensions de la tête du nerf optique. Elle est essentielle pour évaluer les changements structuraux dans le nerf optique.

3. Coloration histologique et immunohistochimie

Coloration histologique : Cette méthode utilise des colorants spécifiques pour mettre en évidence les structures cellulaires et les modifications pathologiques dans les échantillons de tissus. Grâce à cela, on peut directement observer les dommages causés par le glaucome.

Immunohistochimie : En utilisant des anticorps pour détecter des marqueurs spécifiques de la dégénérescence des cellules ganglionnaires rétiniennes, cette technique offre une compréhension approfondie des mécanismes cellulaires impliqués dans la progression du glaucome.

4. Apprentissage automatique et intelligence artificielle

Les algorithmes de machine learning et les réseaux de neurones convolutifs (CNN) sont de plus en plus utilisés pour analyser les images OCT et d'autres types d'images rétiniennes. Ces technologies permettent de détecter des signes précoces de glaucome avec une grande précision en identifiant des motifs subtils souvent invisibles aux cliniciens.

III. Pré-traitement de l'image

1. Elimination des rectangles noirs

Pour certains traitements basés sur la détection des niveaux de gris tel que le seuillage ou le seuillage par hystérésis, il est nécessaire que les objets noirs apparaissant dans certaines images ne soient pas considérés comme des points d'intérêt. Pour cela, nous avons détecté par un algorithme simple ces rectangles, puis nous avons changé leur couleur en un gris moyen. Cette étape de pré-traitement sera effectuée dans l'entièreté des cas et sur l'entièreté des images, même celles n'ayant aucun rectangle de bruit.

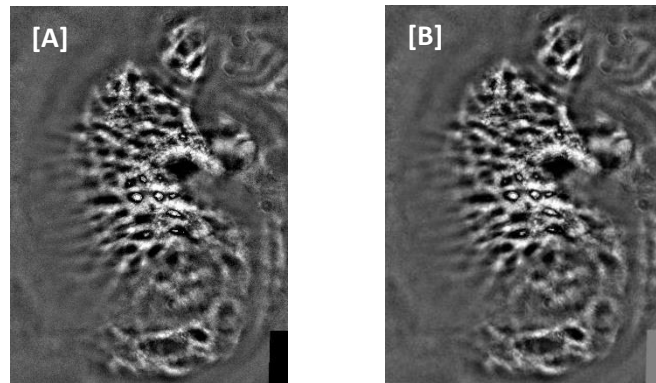


Figure 6 : [A] Image originale de la LC1 [B] LC1 sans le rectangle dérangeant

2. Augmentation du contraste

a. Manipulation à la main

L'augmentation du contraste peut se faire en multipliant tous les pixels de notre image par un facteur de contraste entre 0 et 3. Les facteurs de contraste en dessous de 1 assombrissent l'image tandis que ceux au-dessus de 1 l'éclaircissent.

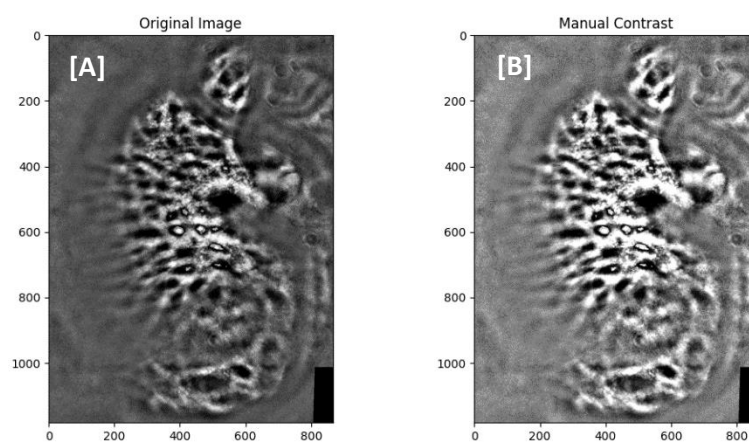


Figure 7: [A] Image initiale de la LC1 [B] Augmentation du contraste sur l'image de la LC1 par modification manuelle (facteur = 1,5)

Cependant, l'inconvénient de cette technique réside dans la diversité des niveaux de gris de chaque image : appliquer le même facteur de contraste à toutes les images pourrait compromettre certaines techniques ultérieures, comme le seuillage.

b. Egalisation d'histogramme

L'augmentation du contraste peut également se faire par égalisation d'histogramme qui consiste à redistribuer les niveaux de gris d'une image par une fonction non linéaire basée sur l'histogramme de l'image. Cela permet d'améliorer le contraste global en étalant les niveaux de gris sur toute la plage de couleurs disponibles. Chaque pixel de l'image est transformé selon la fonction, ce qui uniformise l'histogramme et améliore la visibilité des détails dans les zones sombres et claires.

L'avantage de l'utilisation de l'égalisation d'histogramme dans ce cas précis est dû à la disparité des contrastes et de la luminosité entre chaque image de LC. L'égalisation d'histogramme, étant basée sur l'histogramme initial de l'image et ayant pour but d'étaler l'ensemble des valeurs de 0 à 255, permet d'obtenir des images de LC plus homogènes après traitement.

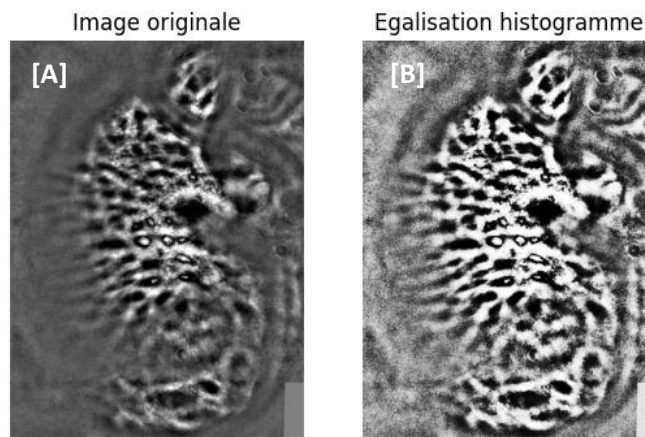


Figure 8: [A] Image initiale de la LC1 [B] Image de la LC1 améliorée par une égalisation d'histogramme

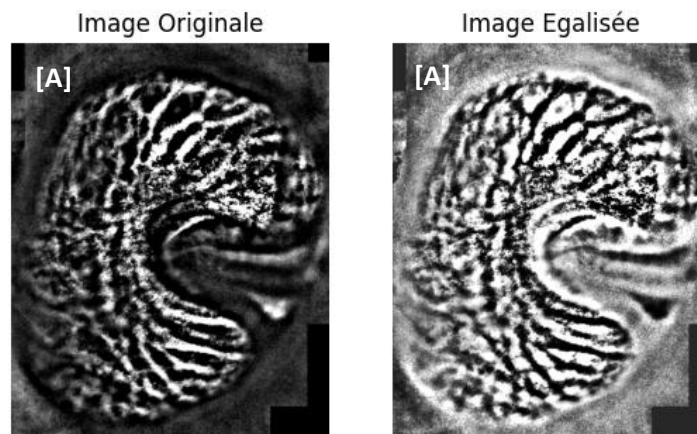


Figure 9: [A] Image initiale de la LC3 [B] Image de la LC3 améliorée par une égalisation d'histogramme

On voit ici clairement le phénomène décrit : les images de la LC1 et de la LC3 ont des tons de couleur initiaux extrêmement différents, ce qui empêcherait, par exemple, de fixer un seuil identique pour filtrer les composants qui nous intéressent. Au contraire, l'image égalisée montre un plus grand contraste et des niveaux de gris plus uniformes. Il devient alors possible d'appliquer des transformations identiques sur les deux images.

c. Top hat et bottom hat

Les opérations top hat et bottom hat sont des techniques de transformation morphologique qui peuvent être utilisées pour améliorer les détails des structures en augmentant les contrastes locaux. Le top hat aide à extraire des éléments brillants sur un fond sombre, mettant en évidence les petites structures claires, tandis le bottom hat aide à extraire des éléments sombres sur un fond clair, mettant en évidence les petites structures sombres.

On fixera la taille de l'élément structurant (SE) de manière à ce qu'elle soit proportionnelle aux structures à détecter : si l'élément est trop petit, il ne détectera pas l'entière des structures ; à l'inverse, s'il est trop grand, il inclura probablement des éléments de fond.

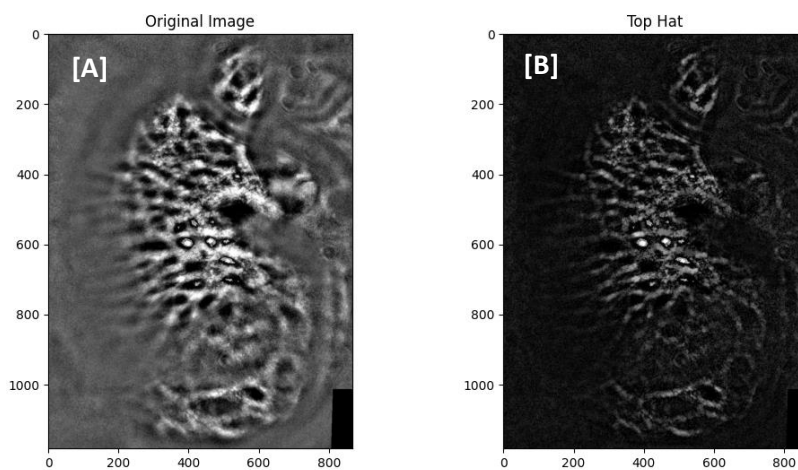


Figure 10: [A] Image initiale de la LC1 [B] Illustration du top hat de l'image de la LC1 pour un SE de taille 12x12

3. Lissage

Une étape de lissage, peut être appliquée pour réduire le bruit et les variations abruptes des niveaux de gris. Pour cela on peut utiliser un filtre gaussien qui applique une fonction gaussienne aux pixels voisins pour produire un effet de flou plus naturel tout en préservant les contours importants. Cette étape de pré-traitement est notamment très utilisée lors de la détection de contour, par exemple avec les noyaux de Sobel ou la méthode de Canny, afin d'améliorer la précision.

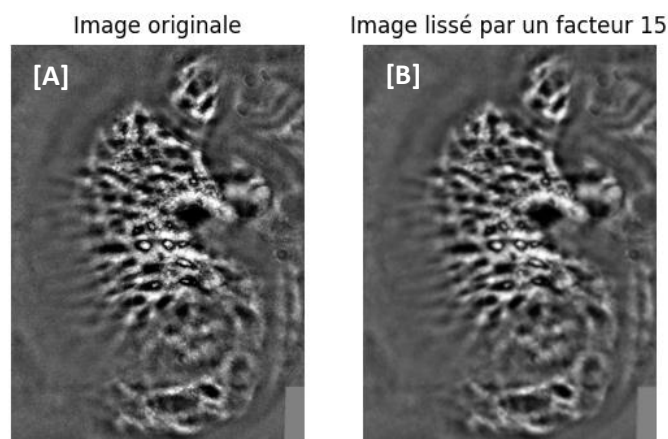


Figure 11: [A] Image initiale de la LC1 [B] Image de la LC1 lissée par un filtre gaussien de facteur 15

IV. Méthodes de segmentation

1. Seuillage

Le seuillage permet de binariser une image et de ne garder que les pixels au-delà d'une certaine valeur en transformant ces pixels en 1 (couleur blanche) et tous les autres en 0 (couleur noire).

Le choix du seuil peut se faire :

- Au hasard (ou par une valeur fixée par l'utilisateur)
- Par la méthode d'Otsu
- Par un seuil Gaussien

a. Seuil fixé par l'utilisateur

Pour une image en niveaux de gris où l'on veut détecter les zones très sombres, on peut définir un seuil proche de 0 : tous les pixels ayant une valeur comprise entre 0 et le seuil deviendront noirs (valeur égale à 0) et tous les pixels supérieurs au seuil deviendront blancs (valeur égale à 1). Étant donné que notre masque de vérité terrain illustre les zones d'intérêt en blanc, il sera préférable d'obtenir l'image inverse de celle créée précédemment. Nous pourrons ainsi comparer notre résultat avec celui attendu.

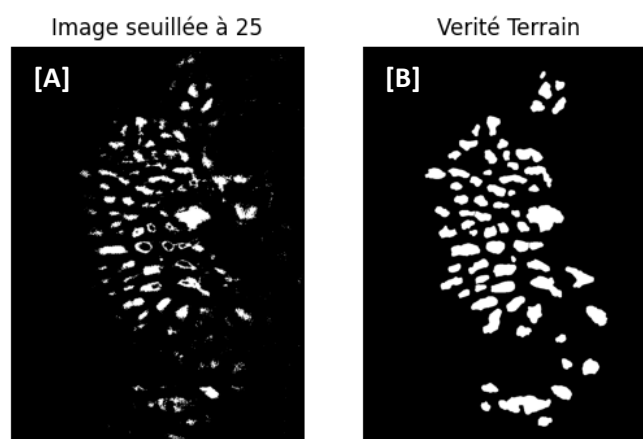


Figure 12: [A] Image de la LC1 ayant subie une élimination des rectangles de bruit, puis étant ayant seuillée à une valeur de 25 [B] Vérité Terrain associé à l'image de la LC1

Naturellement une simple méthode de seuillage ne prend en compte que les couleurs et garde une grande quantité de bruits sur une image non lissée au préalable. C'est pour cela qu'une méthode de seuillage peut être utilisée avec des méthodes de pré-traitement comme du lissage, ou des post-traitements pour diminuer le bruit ou remplir les formes incomplètes.

b. Méthode d'Otsu

La méthode d'Otsu est une méthode de détermination automatique du seuil à partir de l'histogramme de l'image. Elle maximise la variance inter-classe pour séparer l'image en deux classes, permettant ainsi de trouver le seuil optimal pour binariser l'image de manière efficace.

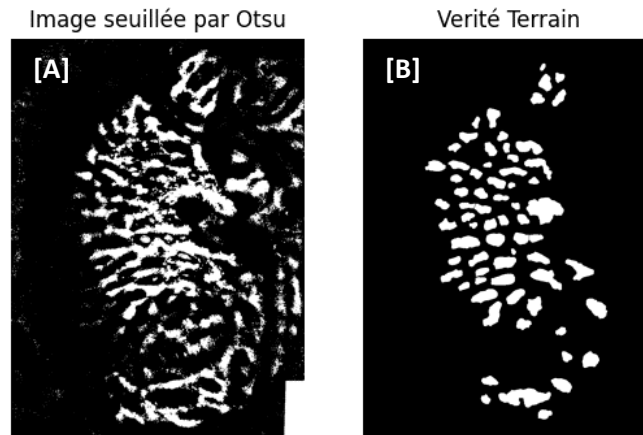


Figure 13: [A] Image de la LC1 ayant subi une élimination des rectangles de bruit, puis ayant été seuillée par la méthode d'Otsu [B] Vérité Terrain associé à l'image de la LC1

Ici malheureusement, on voit nettement que le seuil déterminé par la méthode d'Otsu est trop haut par rapport aux éléments qu'on souhaiterait détecter. On se retrouve donc avec une partie trop importante des structures détectées.

c. Seuil gaussien

La méthode de seuillage par un seuil gaussien consiste à supposer que les niveaux de gris des pixels suivent une distribution gaussienne (normale) et à calculer la moyenne et l'écart-type de cette distribution. On utilise ces mesures pour déterminer un seuil approprié, souvent fixé de telle sorte que qu'il minimise l'erreur de classification entre les classes d'arrière-plan et de premier plan.

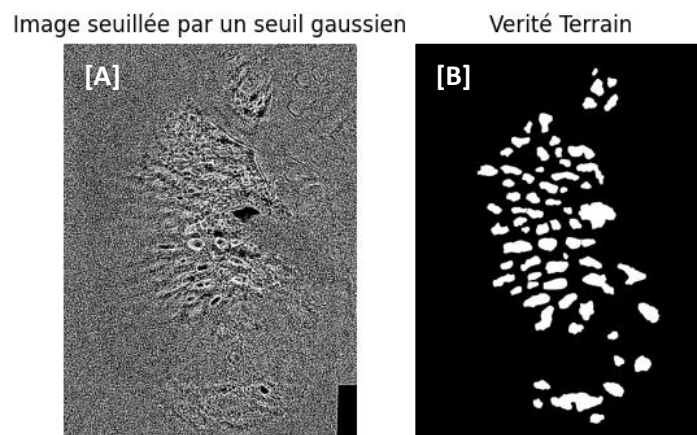


Figure 14: [A] Image de la LC1 ayant subi une élimination des rectangles de bruit, puis ayant été seuillée par un seuil gaussien [B] Vérité Terrain associé à l'image de la LC1

2. Seuillage par hystérésis

Le principe du seuillage par hystérésis est basé sur le choix de deux seuils, avec itération jusqu'à stabilisation. Pour cela, on commence par choisir un seuil bas pour que tous les points d'intérêt soient détectés, et un seuil haut pour que seuls les points d'intérêt soient détectés. Tous les pixels supérieurs au seuil haut sont étiquetés comme « points d'intérêt », et pour tous les autres, s'ils sont situés entre les deux seuils et qu'ils ont dans leur voisinage un point d'intérêt, ils sont également étiquetés comme « points d'intérêt ». L'algorithme est itéré jusqu'à ce qu'aucun nouveau point d'intérêt puisse être étiqueté.

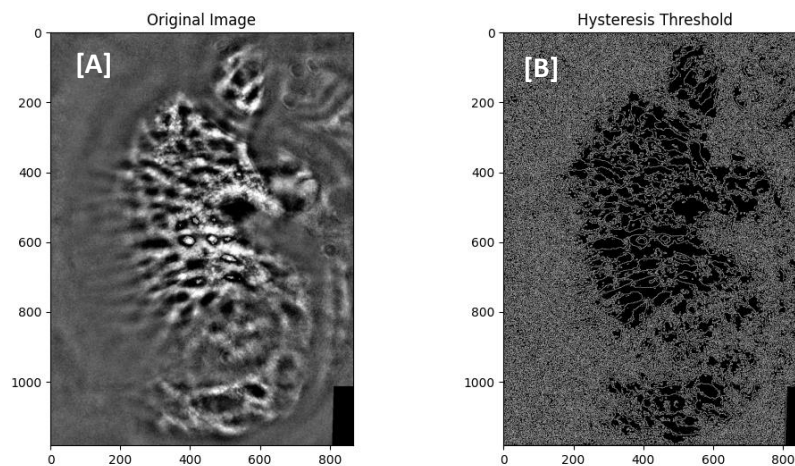


Figure 15: [A] Image initiale de la LC1 [B] Représentation de l'image de la LC1 après un seuillage par hystérésis

3. Détection des contours

Les techniques de détection de contours permettent de renforcer les limites des structures d'intérêt, facilitant ainsi la segmentation. Parmi les méthodes utilisées, on trouve :

- **Filtre de Sobel** : Méthode utilisée pour détecter les gradients d'intensité dans l'image, ce filtre met en évidence les bords.
- **Filtre de Canny** : Méthode efficace pour la détection des contours qui détecte les bords en utilisant une approche de double seuil.

a. Noyaux de Sobel

Le filtre de Sobel est une méthode de détection de contours qui utilise des dérivées discrètes pour calculer les gradients d'intensité dans l'image. Cela permet de mettre en évidence les régions où l'intensité des pixels change brusquement, ce qui correspond souvent aux bords des objets dans l'image. Le filtre de Sobel applique deux convolutions séparées pour détecter les gradients horizontaux (G_x) et verticaux (G_y), puis combine ces gradients pour obtenir une image finale des contours.

Les contours détectés sont principalement les gradients forts des structures dans l'image. Cependant, le filtre de Sobel peut être sensible au bruit et aux petits détails non pertinents.

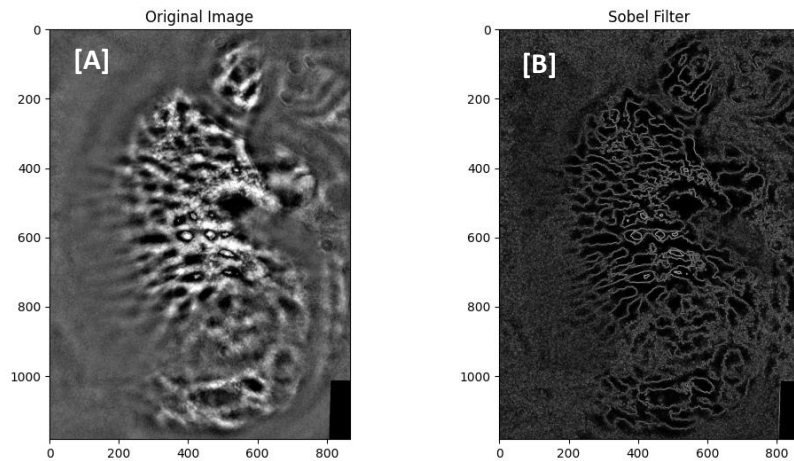


Figure 16: [A] Image initiale de la LC1 [B] Représentation de l'image de la LC1 après l'application de noyaux de Sobel

b. Filtre de Canny

Le filtre de Canny est une méthode plus sophistiquée de détection de contours qui fonctionne en plusieurs étapes :

- 1) **Lissage gaussien** : L'image est lissée pour réduire le bruit.
- 2) **Calcul des gradients** : Comme avec le filtre de Sobel, les gradients horizontaux et verticaux sont calculés.
- 3) **Suppression des non-maxima** : Seuls les pixels les plus forts (correspondant aux bords les plus nets) sont conservés.
- 4) **Double seuillage** : Deux seuils sont appliqués pour distinguer les bords forts, les bords faibles et les non-bords.
- 5) **Hystérésis** : Les bords faibles sont inclus dans le résultat final uniquement s'ils sont connectés à des bords forts.

Les contours sont plus nets et bien définis par rapport à ceux obtenus avec le filtre de Sobel. La méthode de double seuillage et d'hystérésis permet de réduire les faux positifs dus au bruit, ce qui donne des contours plus propres et mieux connectés.

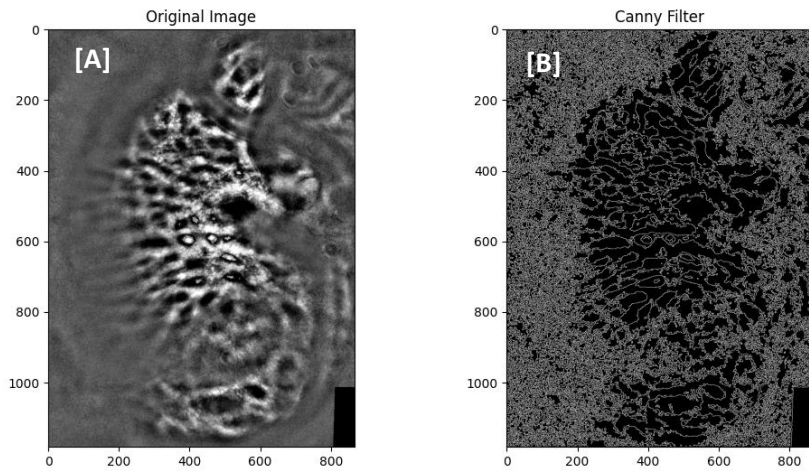


Figure 17: [A] Image initiale de la LC1 [B] Représentation de l'image de la LC1 après l'application du filtre de Canny

V. Amélioration de l'image

1. Ouverture et Fermetures

Les opérations d'ouverture et de fermeture sont des techniques de morphologie mathématique utilisées pour améliorer les résultats de segmentation en éliminant le bruit et en comblant les petites lacunes. Ces opérations sont applicables aussi bien sur des images en niveaux de gris que sur des images binaires et prennent en paramètre un élément structurant (SE) d'une certaine taille.

- **Ouverture** : Une opération qui consiste en une érosion suivie d'une dilatation. Elle est utile pour supprimer les petits objets blancs présents dans l'image.
- **Fermeture** : Une opération qui consiste en une dilatation suivie d'une érosion. Elle est utile pour combler les petits trous noirs ou espaces présents dans les objets de l'image.

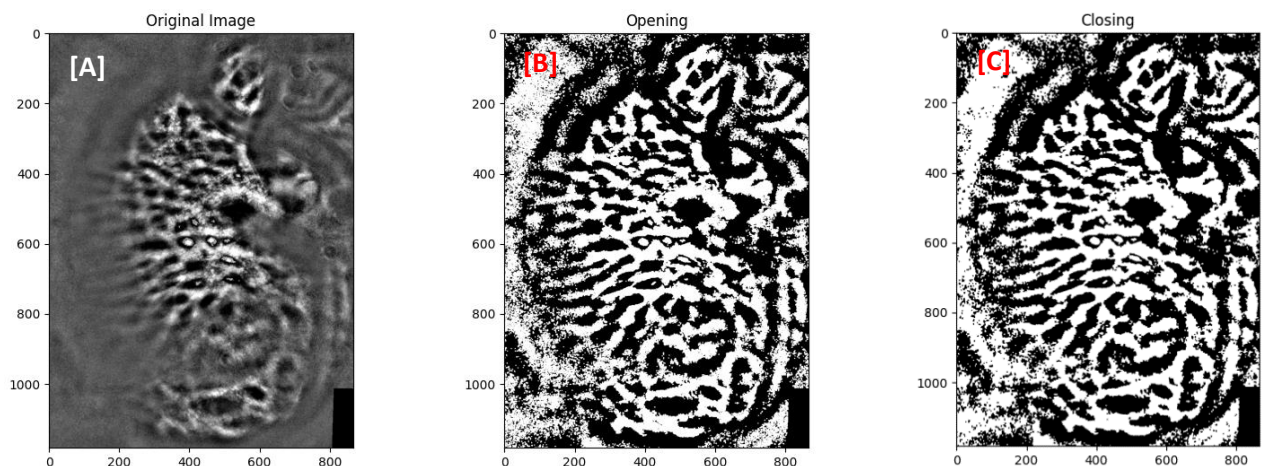


Figure 18: [A] Image initiale de la LC1 [B] Ouverture appliquée à l'image en niveaux de gris de la LC1 avec un SE de taille 3x3 [C] Fermeture appliquée à l'image en niveaux de gris de la LC1 avec un SE de taille 3x3

2. Filtres alternés séquentiels

Les filtres séquentiels alternés consistent en une série d'opérations d'ouverture et de fermeture appliquées successivement. Cette technique permet d'améliorer la segmentation en réduisant le bruit tout en préservant les structures importantes en alternant ouverture, qui élimine le bruit et fermeture, qui remplit les trous et connecte les objets disjoints.

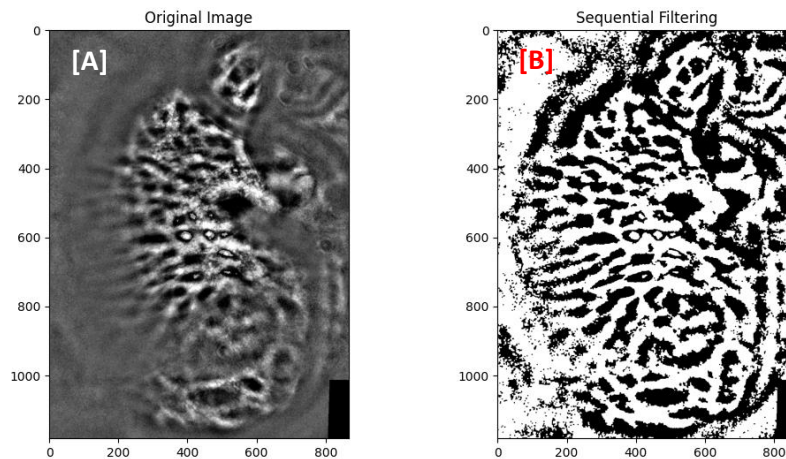


Figure 19: [A] Image initiale de la LC1 [B] Filtres alternés séquentiels (2 itérations) appliqués à l'image en niveaux de gris de la LC1 avec des SE de taille 3 et 4

3. Comblers les trous

Afin de combler les trous noirs pouvant être présent au centre des structures fermées blanches, il est possible d'utiliser un algorithme de rebouchage de trou. Ce genre d'algorithme peut être codé de plusieurs façons, mais il s'agit dans le cas présenté ci-contre, d'utiliser une technique de reconstruction par dilatation à partir des bords extérieurs de l'image.

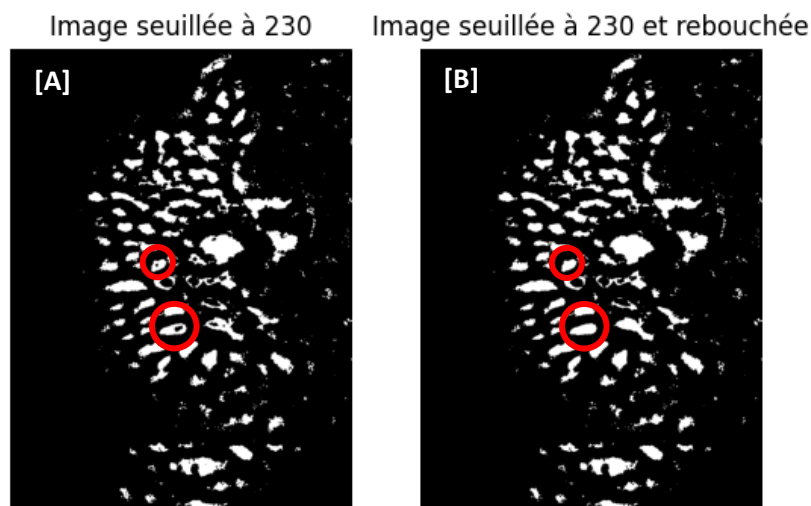


Figure 20: [A] Image de la LC1 après égalisation d'histogramme et seuillage à 230 [B] Image de la LC1 après égalisation d'histogramme, seuillage à 230, et remplissage des trous

VI. Expérimentation

Après avoir étudié certaines des méthodes de traitement d'image existantes, nous pouvons assembler plusieurs étapes de pré-traitement, de segmentation et d'amélioration d'image pour essayer d'obtenir le masque binaire le plus proche possible de notre vérité terrain. Pour cela, il s'agit d'exécuter les différentes fonctions dans un ordre bien précis et de mesurer la **précision**, le **rappel** et le **F1 score** de chaque image par rapport à sa vérité terrain. Pour rappel, voici à quoi correspondent ces métriques :

- Vrai Positifs (VP) : le pixel vaut 1 dans l'image et dans la vérité terrain
- Faux Positifs (FP) : le pixel vaut 1 dans l'image mais 0 dans la vérité terrain
- Vrais Négatifs (VN) : le pixel vaut 0 dans l'image et dans la vérité terrain
- Faux négatifs (FN) : le pixel vaut 0 dans l'image mais 1 dans la vérité terrain
- Précision (P) : c'est la proportion d'items pertinents parmi l'ensemble des items proposés, elle mesure la qualité des prédictions positives du modèle.

$$P = \frac{VP}{VP + FP}$$

- Rappel (R) : c'est la proportion d'items pertinents proposés parmi l'ensemble des items pertinents, il mesure la capacité du modèle à identifier correctement toutes les instances positives.

$$R = \frac{VP}{VP + FN}$$

Étant donné les différents échantillons d'images disponibles, il est important de ne pas évaluer nos résultats uniquement sur une image de LC spécifique, mais sur plusieurs, afin de s'assurer de la compatibilité de nos méthodes sur toutes les images.

Voici plusieurs expériences de traitement d'image qui ont été menées dans le but d'obtenir des résultats se rapprochant le plus possible de notre masque de vérité terrain :

- Méthode de seuillage
- Méthode d'ouverture et fermeture
- Méthode de Top Hat

1. Méthode de seuillage

Une première méthode basée sur un seuillage simple avec choix manuel du seuil fut testée. Les traitements appliqués sur l'image sont :

- 1) **L'élimination des rectangles noirs** pour éviter qu'ils soient détectés comme points d'intérêts.
- 2) **Inversion de l'image** pour que les points d'intérêts apparaissent non pas avec une valeur de 0 mais avec une valeur de 1.
- 3) **Egalisation d'histogramme** pour que toutes les images puissent subir les mêmes modifications en espérant obtenir des qualités de résultats similaire aussi.
- 4) **Application d'un seuil à 230**, valeur trouvée par un algorithme qui testait les valeurs jusqu'à trouver le meilleur score F1.
- 5) **Remplissage des trous** pour se rapprocher au maximum de la vérité terrain.
- 6) **Détection et approximation des contours** par une fonction qui parcourt l'image pour trouver des groupes de pixels blancs, puis qui part de point de contour détecté, suit le contour en suivant les pixels adjacents qui sont également blancs et ce jusqu'à ce que le contour complet soit détecté. La fonction ne garde que les formes qui possèdent une aire plus grande qu'une valeur de seuil déterminée au cours de l'algorithme de maximisation du score F1.

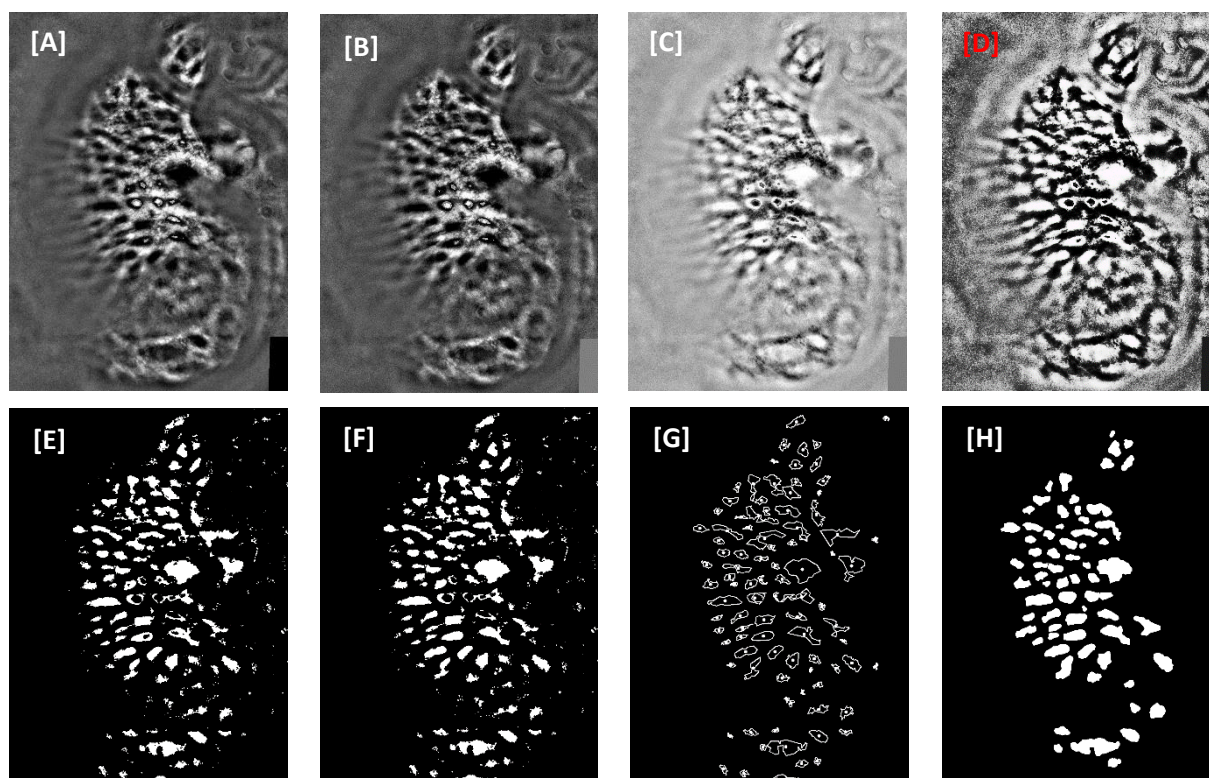


Figure 21: Image de la LC1 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application du seuil [F] l'étape 5 du remplissage des trous [G] l'étape 6 de la détection de contour et des calculs de quantifications [H] la vérité terrain à obtenir

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[G]	0,7218	0,6523	0,6853	84 (+24%)	838	768
[H]				68	1202	933

On aperçoit ici que la détection des contours et l'idée de ne garder que les structures plus grandes qu'un certain seuil permettent d'éliminer beaucoup de bruit. On se retrouve dans ce cas ça avec un résultat assez satisfaisant, mais qui possède un certain nombre de structures en surnombre au niveau du bord de la lame criblée.

Afin de savoir si cette technique pourrait être utilisé sur l'entièreté des autres images, on va l'appliquer sur une des images les plus différentes et plus compliqué de notre base de données : la LC3.

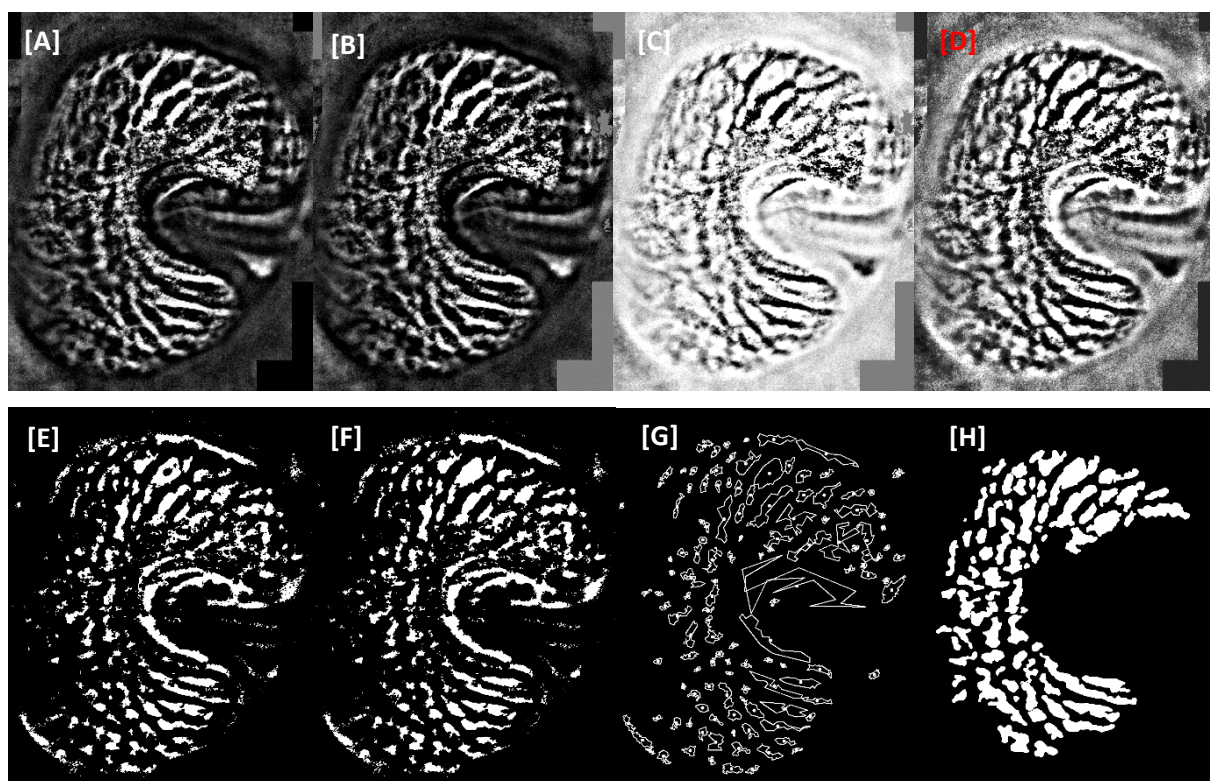


Figure 22: Image de la LC3 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application du seuil [F] l'étape 5 du remplissage des trous [G] l'étape 6 de la détection de contour et des calculs de quantifications [H] la vérité terrain à obtenir

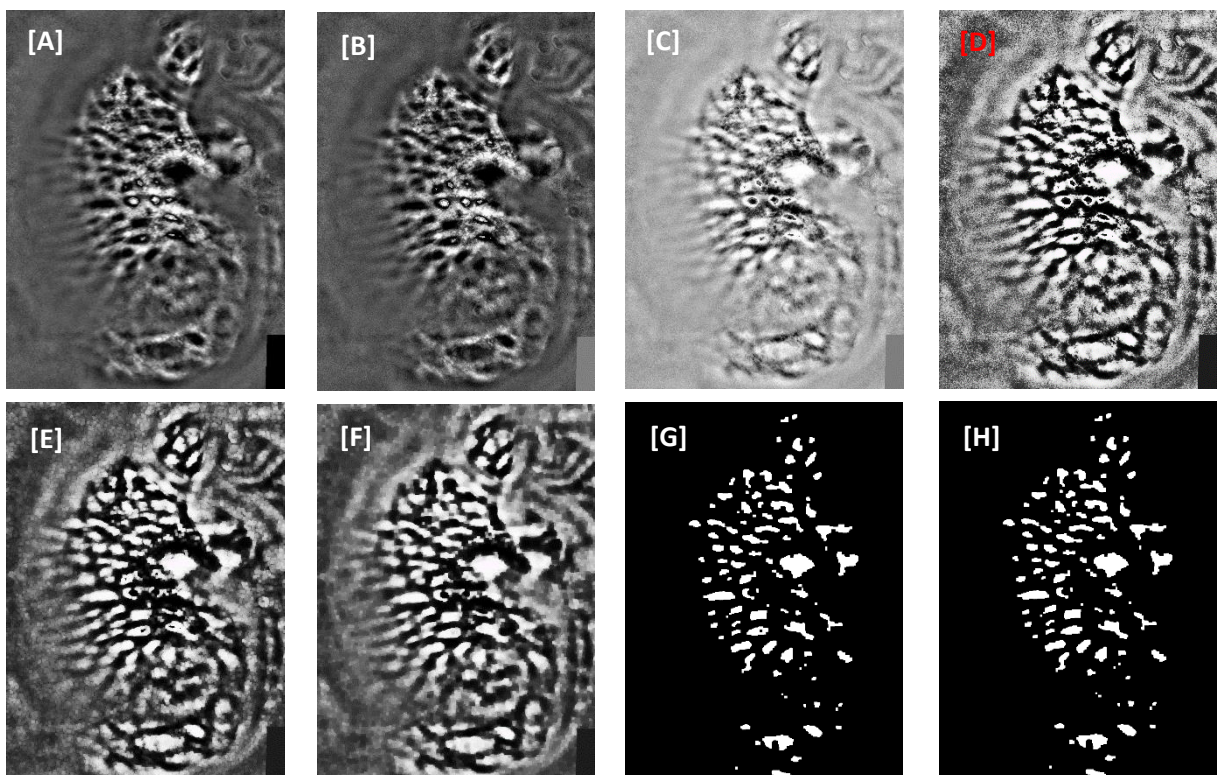
	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[G]	0,5726	0,4641	0,5127	125 (+68%)	923	1536
[H]				74	1997	1719

L'image de la LC3 qui a de nettement moins bons résultats de quantification que la LC1: les valeurs de précision et de rappel ne sont d'autour de 0,5 ce qui veut dire qu'il y a beaucoup trop de faux positifs. On le voit notamment au niveau du nombre de formes détectées dans l'image qui est 1,68 fois supérieur au nombre de forme qu'on est censé détecter. Voyons s'il est possible d'obtenir de meilleurs résultats avec d'autres techniques.

2. Méthode d'ouverture et fermeture

Une deuxième méthode basée sur l'application d'une ouverture et d'une fermeture avant l'application d'un seuillage fut testée. Les traitements appliqués sur l'image sont :

- 1) **L'élimination des rectangles noirs** pour éviter qu'ils soient détectés comme points d'intérêts.
- 2) **Inversion de l'image** pour que les points d'intérêts apparissent non pas avec une valeur de 0 mais avec une valeur de 1.
- 3) **Egalisation d'histogramme** pour que toutes les images puissent subir les mêmes modifications en espérant obtenir des qualités de résultats similaire aussi.
- 4) **Ouverture** avec un SE de taille 7x7 choisi après plusieurs tests manuels, utile pour supprimer le bruit dans le fond de l'image.
- 5) **Fermeture** avec un SE de taille 9x9 choisi après plusieurs tests manuels, pour boucher les petits trous ou souder des objets.
- 6) **Application d'un seuil à 230**, valeur trouvée par un algorithme qui testait les valeurs jusqu'à trouver le meilleur score F1.
- 7) **Remplissage des trous** pour se rapprocher au maximum de la vérité terrain.
- 8) **Détection et approximation des contours** par une fonction qui parcourt l'image pour trouver des groupes de pixels blancs, puis qui part de point de contour détecté, suit le contour en suivant les pixels adjacents qui sont également blancs et ce jusqu'à ce que le contour complet soit détecté. La fonction ne garde que les formes qui possèdent une aire plus grande qu'une valeur de seuil déterminée au cours de l'algorithme de maximisation du score F1.



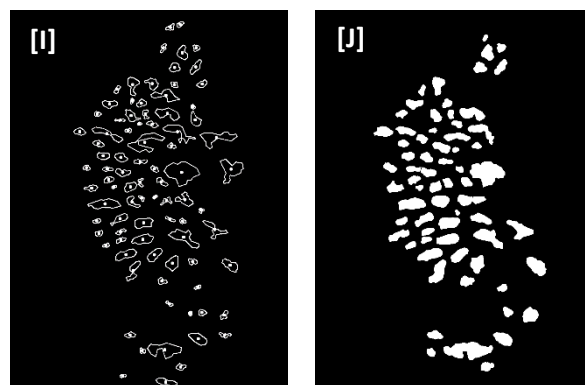
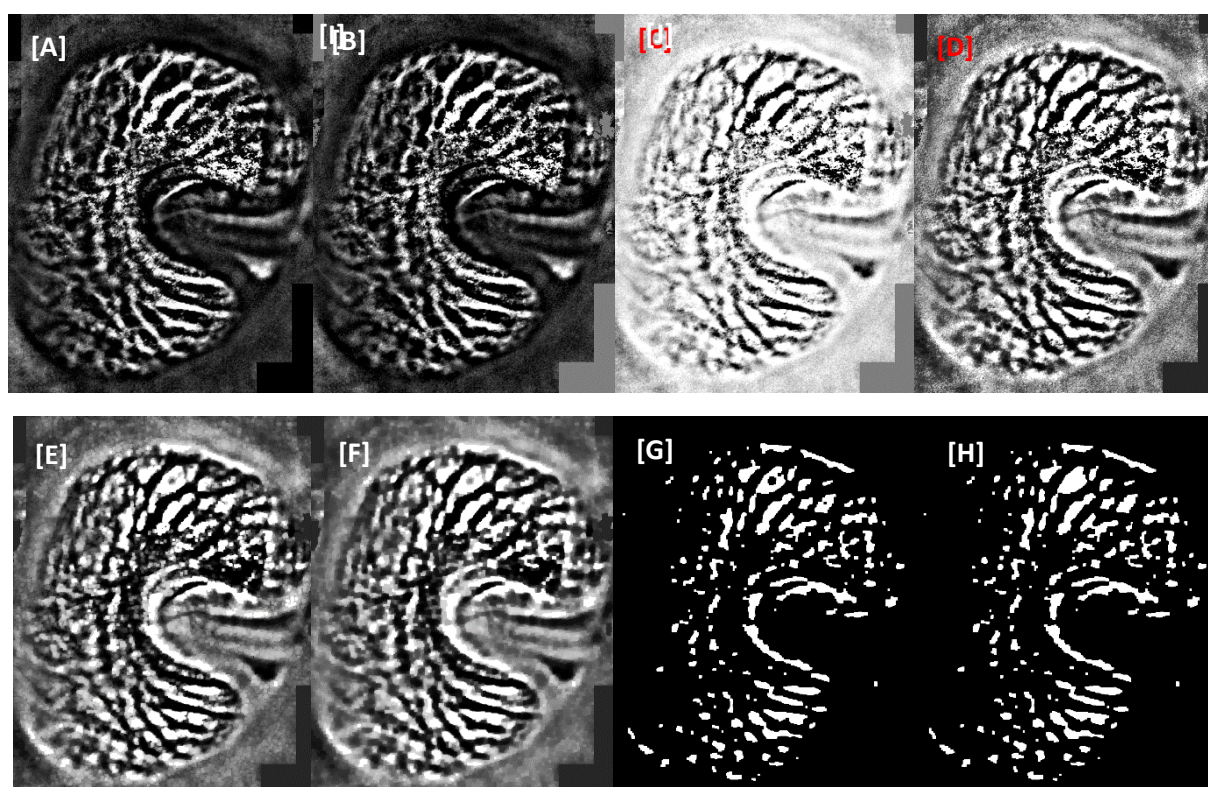


Figure 23: Image de la LC1 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application de l'ouverture avec un SE de taille 71 [F] l'étape 5 de l'application de la fermeture avec un SE de taille 9 [G] l'étape 6 l'application d'un seuil à 230 [H] l'étape 7 du remplissage des trous [I] l'étape 8 de la détection de contour et des calculs de quantifications [J] la vérité terrain à obtenir

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[I]	0,7712	0,6162	0,6850	87 (+28%)	736	775
[J]				68	1202	933

Cette fois-ci, l'image obtenue possède moins de petites structures additionnelles de bruit sur le contour de la lame criblée et se présente donc avec une meilleure précision. Mais le manque de détection de ces pixels implique que moins de pixel ont été détectés y compris sur les plus grosse structure, diminuant ainsi le rappel. Comparons encore une fois avec une image de lame criblée très différente de LC1. Pour pouvoir continuer nos comparaisons, nous choisirons LC3.



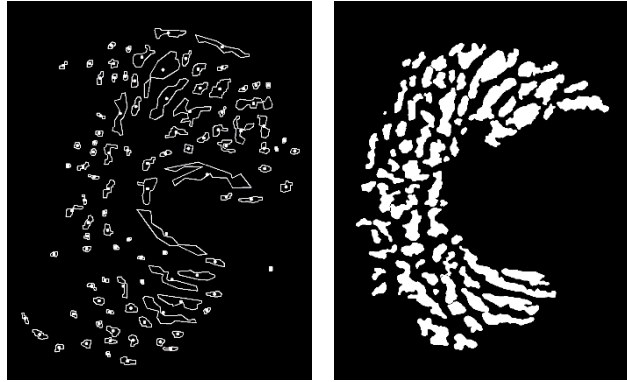


Figure 24: Image de la LC1 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application de l'ouverture avec un SE de taille 71 [F] l'étape 5 de l'application de la fermeture avec un SE de taille 9 [G] l'étape 6 l'application d'un seuil à 230 [H] l'étape 7 du remplissage des trous [I] l'étape 8 de la détection de contour et des calculs de quantifications [J] la vérité terrain à obtenir

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[I]	0,6086	0,4074	0,4881	103 (+39%)	943	1202
[J]				74	1997	1719

Cette fois-ci le nombre de forme n'est pas 1,68 fois que qu'il est censé être mais seulement 1,39 fois ! Malheureusement les formes qui n'ont pas été détectés sont des zones d'intérêts ce qui fausse nos résultats. On se retrouve donc avec un taux de rappel de 0,4 ce qui est très faible, cette technique est donc délétère pour notre image de la CT3, et ne sera pas une technique à garder.

Passons donc à une troisième méthode en espérant qu'elle soit plus performante.

3. Méthode des Top Hat

Cette troisième méthode consiste à appliquer des modifications sur l'image afin de faire apparaître les points d'intérêts avant d'effectuer le seuillage. Les traitements appliqués sur l'image sont :

- 1) **L'élimination des rectangles noirs** pour éviter qu'ils soient détectés comme points d'intérêts.
- 2) **Inversion de l'image** pour que les points d'intérêts apparissent non pas avec une valeur de 0 mais avec une valeur de 1.
- 3) **Egalisation d'histogramme** pour que toutes les images puissent subir les mêmes modifications en espérant obtenir des qualités de résultats similaire aussi.
- 4) **Appliquer un TopHat** afin de faire ressortir les structures blanches donc les structures d'intérêt. Pour cela on a besoin d'une SE qu'on va fixer à une valeur de 59 qui nous a été donné par un programme cherchant la taille de SE correspondant au plus grand score f1.
- 5) **Application d'un seuil à 230**, valeur trouvée par un algorithme qui testait les valeurs jusqu'à trouver le meilleur score F1.
- 6) **Remplissage des trous** pour se rapprocher au maximum de la vérité terrain.
- 7) **Détection et approximation des contours** par une fonction qui parcourt l'image pour trouver des groupes de pixels blancs, puis qui part de point de contour détecté, suit le contour en suivant les pixels adjacents qui sont également blancs et ce jusqu'à ce que le contour complet soit détecté. La fonction ne garde que les formes qui possèdent une aire plus grande qu'une valeur de seuil déterminée au cours de l'algorithme de maximisation du score F1.

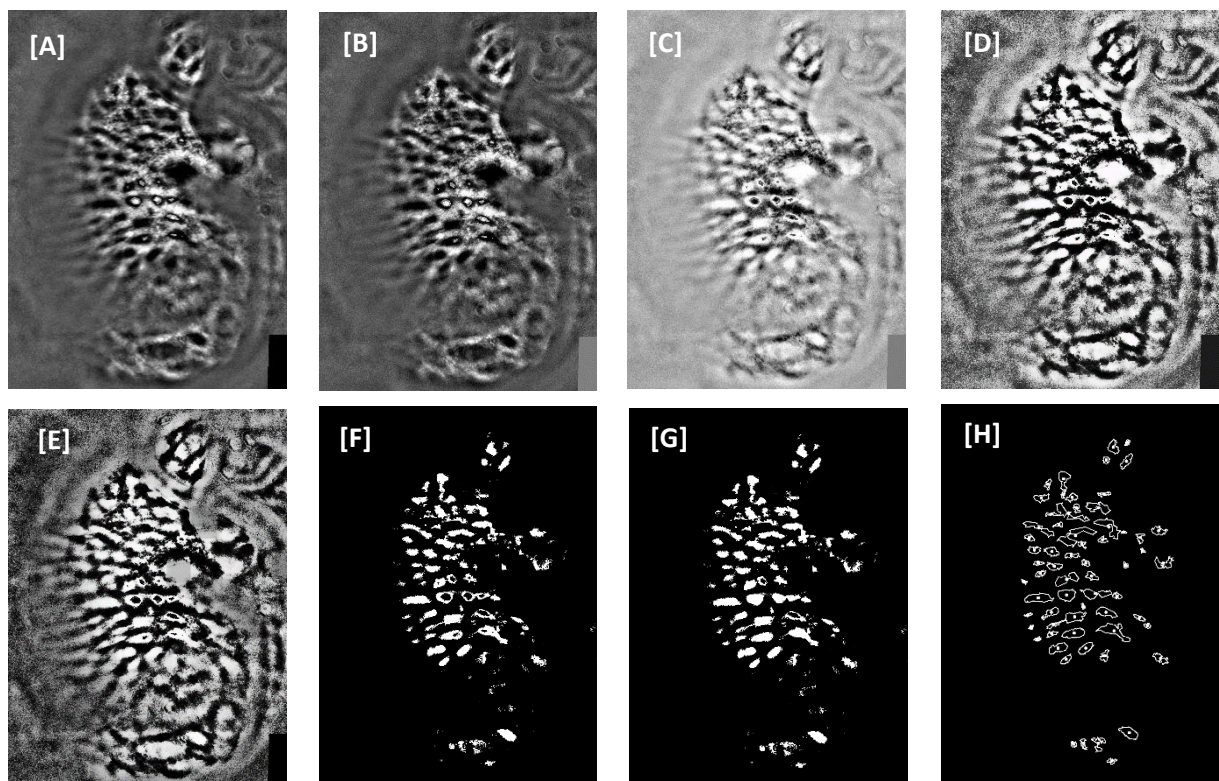


Figure 25 : Image de la LC1 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application du top hat [F] l'étape 5 l'application d'un seuil à 230 [G] l'étape 6 du remplissage des trous [H] l'étape 7 de la détection de contour et des calculs de quantifications [I] la vérité terrain à obtenir

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[H]	0,8850	0,4896	0,6304	64 (-6%)	658	526
[I]				68	1202	933

Nous avons ici un résultat totalement différent de précédemment, avec une grande précision au dépend du rappel. Mis ensemble le score f1 est inférieur à ceux trouver précédemment, mais cette image a la particularité d'avoir détecter moins de structures qu'il n'y en a réellement. Cela provient du fait que les structures se trouvant autour de la lame criblée n'ont pas été détectée, mais en conséquence les structures détectées même celle existante, sont beaucoup plus petite. Il pourrait s'agir une bonne façon d'obtenir le centre de chaque structure d'intérêt, et de reconstruire cette structure par la suite grâce au reste de nos informations. Mais vérifions ce que fait cet algorithme sur notre image de la LC3.

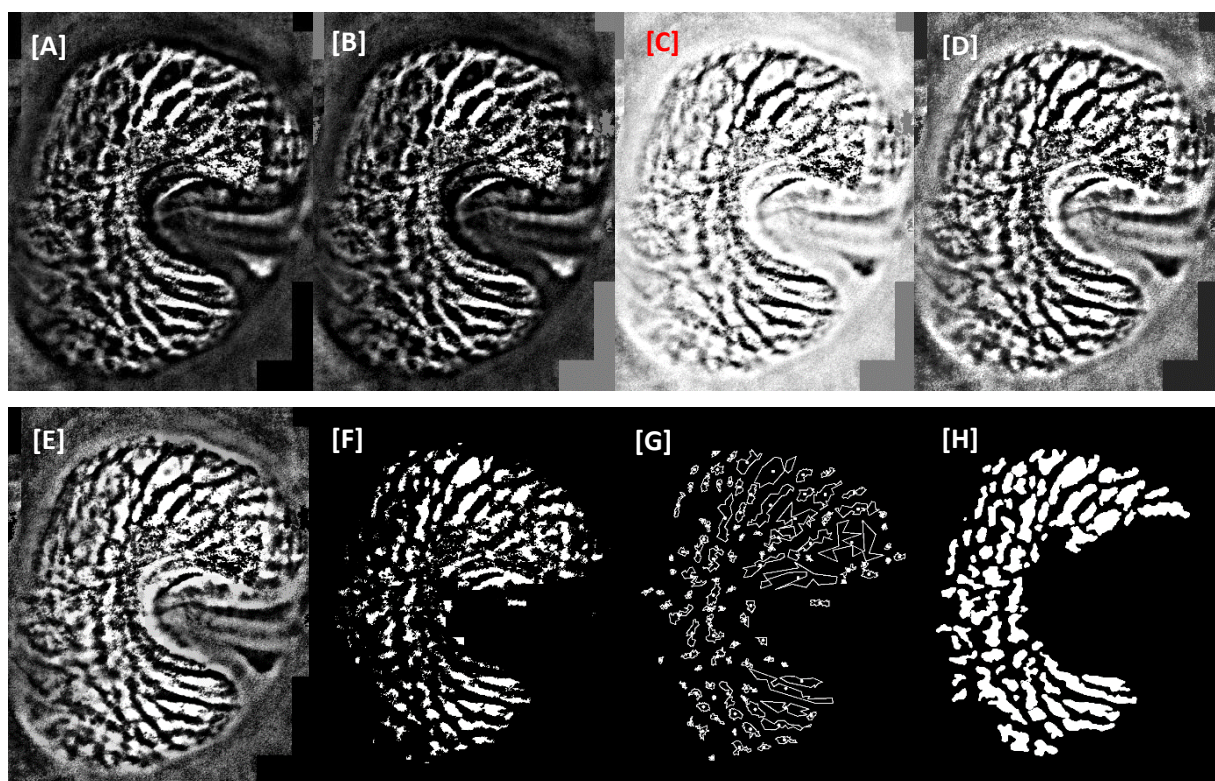


Figure 26: Image de la LC3 dans [A] son état original [B] l'étape 1 d'éclaircissement des rectangles des bords [C] l'étape 2 d'inversion des niveaux de gris [D] l'étape 3 de l'égalisation d'histogramme [E] l'étape 4 de l'application du top hat [F] l'étape 5 l'application d'un seuil à 230 [G] l'étape 7 de la détection de contour et des calculs de quantifications [H] la vérité terrain à obtenir

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
[G]	0,7327	0,4940	0,5902	106 (+ 43%)	878	1256
[H]				74	1997	1719

Les résultats pour cette image sont étonnement bon par rapport aux précédents. En effet, on obtient un score F1 de presque 0,6 ce qui est convenable. On remarque alors que le contour de la lame criblée a moins été sélectionné comme structure d'intérêt ce qui est une bonne chose, et que notre taux de précision est donc assez haut. Un grand nombre de pixel manque à l'appel, mais un grand nombre d'entre eux sont ceux qu'on ne souhaitait pas garder.

IV. Comparaison

Pour comparer les résultats obtenus pour chaque, on les a disposés dans un même tableau :

	Précision	Rappel	F1 score	Nb de formes	Aire moyenne	Ecart-type aire
Im1				68	1202	933
Im3				74	1997	1719
Im1 Seuil	0,7218	0,6523	0,6853	84 (+24%)	838	768
Im1 Ouv	0,7712	0,6162	0,6850	87 (+28%)	736	775
Im1 Tophat	0,8850	0,4896	0,6304	64 (-6%)	658	526
Im3 Seuil	0,5726	0,4641	0,5127	125 (+68%)	923	1536
Im3 Ouv	0,6086	0,4074	0,4881	103 (+39%)	943	1202
Im3 Tophat	0,7327	0,4940	0,5902	106 (+ 43%)	878	1256

Même si la meilleure technique pour l'analyse de l'image de la LC1 n'est pas certaine et dépend de ce que souhaite l'utilisateur, il est assez net que la technique 3 de Top hat fonctionne bien mieux que les autres sur l'image de la LC3. Au vu de nos résultats, on peut se douter qu'il doit être possible de faire mieux en combinant plusieurs techniques, mais en attendant la technique numéro 3 semble plus apte à s'adapter peu importe l'image et le contraste de la lame criblée.

VII. Intelligence Artificielle

Enfin, nous avons tenté d'utiliser des techniques d'intelligence artificielle pour améliorer nos résultats. Dans tous nos modèles, nous avons voulu minimiser la mean square error (MSE).

1. Convolutional Neural Network

Pour ce modèle, nous avons utilisé les images originales sans prétraitements. L'ensemble d'entraînement était constitué de 7 images de LC et des 7 masques associés, tandis que l'ensemble de test comprenait 3 images de LC et les 3 masques associés.

Les résultats obtenus avec ce modèle n'ont pas été satisfaisants du tout. En effet, nous avons fait face à des grandes contraintes. Premièrement, la limitation de la mémoire vive (RAM) disponible a réduit notre capacité à réaliser un nombre suffisant d'époques d'entraînement. Deuxièmement, le manque de données a également compromis la capacité du modèle à apprendre efficacement, ce qui a résulté en des performances très mauvaises.

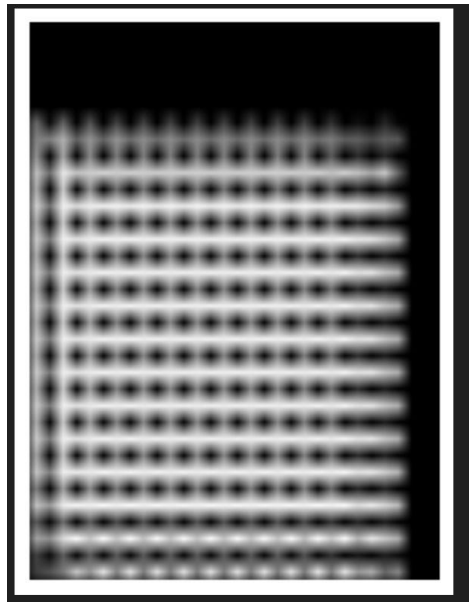


Figure 27: Résultat obtenu avec le CNN

2. U-Net

Pour notre second modèle, nous avons opté pour un U-Net. Son architecture est la suivante :

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 768, 512, 5)	0	-
conv2d (Conv2D)	(None, 384, 256, 32)	896	input_layer[0][0]
batch_normalization (BatchNormalization)	(None, 384, 256, 32)	128	conv2d[0][0]
activation (Activation)	(None, 384, 256, 32)	0	batch_normalization[0][0]
activation_1 (Activation)	(None, 384, 256, 32)	0	activation[0][0]
separable_conv2d (SeparableConv2D)	(None, 384, 256, 64)	2,400	activation_1[0][0]
batch_normalization (BatchNormalization)	(None, 384, 256, 64)	256	separable_conv2d[0][0]
activation_2 (Activation)	(None, 384, 256, 64)	0	batch_normalization[0][0]
separable_conv2d_1 (SeparableConv2D)	(None, 384, 256, 64)	4,736	activation_2[0][0]
batch_normalization (BatchNormalization)	(None, 384, 256, 64)	256	separable_conv2d_1[0][0]
max_pooling2d (MaxPooling2D)	(None, 192, 128, 64)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 192, 128, 64)	2,112	activation[0][0]
add (Add)	(None, 192, 128, 64)	0	max_pooling2d[0][0]
activation_3 (Activation)	(None, 192, 128, 64)	0	add[0][0]
separable_conv2d_2 (SeparableConv2D)	(None, 192, 128, 128)	8,896	activation_3[0][0]
batch_normalization (BatchNormalization)	(None, 192, 128, 128)	512	separable_conv2d_2[0][0]
activation_4 (Activation)	(None, 192, 128, 128)	0	batch_normalization[0][0]
separable_conv2d_3 (SeparableConv2D)	(None, 192, 128, 128)	17,664	activation_4[0][0]
batch_normalization (BatchNormalization)	(None, 192, 128, 128)	512	separable_conv2d_3[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 96, 64, 128)	0	batch_normalization[0][0]
conv2d_2 (Conv2D)	(None, 96, 64, 128)	8,320	add[0][0]
add_1 (Add)	(None, 96, 64, 128)	0	max_pooling2d_1[0][0]
activation_5 (Activation)	(None, 96, 64, 128)	0	add_1[0][0]
separable_conv2d_4 (SeparableConv2D)	(None, 96, 64, 256)	34,376	activation_5[0][0]
batch_normalization (BatchNormalization)	(None, 96, 64, 256)	1,024	separable_conv2d_4[0][0]
activation_6 (Activation)	(None, 96, 64, 256)	0	batch_normalization[0][0]
separable_conv2d_5 (SeparableConv2D)	(None, 96, 64, 256)	68,896	activation_6[0][0]
batch_normalization (BatchNormalization)	(None, 96, 64, 256)	1,024	separable_conv2d_5[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 48, 32, 256)	0	batch_normalization[0][0]
conv2d_3 (Conv2D)	(None, 48, 32, 256)	33,024	add_1[0][0]
add_2 (Add)	(None, 48, 32, 256)	0	max_pooling2d_2[0][0]
activation_7 (Activation)	(None, 48, 32, 256)	0	add_2[0][0]
conv2d_transpose (Conv2DTranspose)	(None, 48, 32, 256)	590,880	activation_7[0][0]
batch_normalization (BatchNormalization)	(None, 48, 32, 256)	1,024	conv2d_transpose[0][0]
activation_8 (Activation)	(None, 48, 32, 256)	0	batch_normalization[0][0]
conv2d_transpose_1 (Conv2DTranspose)	(None, 48, 32, 256)	590,880	activation_8[0][0]
batch_normalization (BatchNormalization)	(None, 48, 32, 256)	1,024	conv2d_transpose_1[0][0]
up_sampling2d_1 (UpSampling2D)	(None, 96, 64, 256)	0	add_2[0][0]
up_sampling2d (UpSampling2D)	(None, 96, 64, 256)	0	batch_normalization[0][0]
conv2d_4 (Conv2D)	(None, 96, 64, 256)	65,792	up_sampling2d_1[0][0]
add_3 (Add)	(None, 96, 64, 256)	0	up_sampling2d[0][0]
activation_9 (Activation)	(None, 96, 64, 256)	0	add_3[0][0]
conv2d_transpose_2 (Conv2DTranspose)	(None, 96, 64, 128)	295,840	activation_9[0][0]
batch_normalization (BatchNormalization)	(None, 96, 64, 128)	512	conv2d_transpose_2[0][0]
activation_10 (Activation)	(None, 96, 64, 128)	0	batch_normalization[0][0]
conv2d_transpose_3 (Conv2DTranspose)	(None, 96, 64, 128)	147,584	activation_10[0][0]
batch_normalization (BatchNormalization)	(None, 96, 64, 128)	512	conv2d_transpose_3[0][0]
up_sampling2d_3 (UpSampling2D)	(None, 192, 128, 256)	0	add_3[0][0]
up_sampling2d_2 (UpSampling2D)	(None, 192, 128, 128)	0	batch_normalization[0][0]
conv2d_5 (Conv2D)	(None, 192, 128, 128)	32,896	up_sampling2d_3[0][0]
add_4 (Add)	(None, 192, 128, 128)	0	up_sampling2d_2[0][0]
activation_11 (Activation)	(None, 192, 128, 128)	0	add_4[0][0]
conv2d_transpose_4 (Conv2DTranspose)	(None, 192, 128, 64)	73,792	activation_11[0][0]
batch_normalization (BatchNormalization)	(None, 192, 128, 64)	256	conv2d_transpose_4[0][0]
activation_12 (Activation)	(None, 192, 128, 64)	0	batch_normalization[0][0]
conv2d_transpose_5 (Conv2DTranspose)	(None, 192, 128, 64)	36,928	activation_12[0][0]
up_sampling2d_5 (UpSampling2D)	(None, 384, 256, 128)	0	add_4[0][0]
up_sampling2d_4 (UpSampling2D)	(None, 384, 256, 64)	0	batch_normalization[0][0]
conv2d_6 (Conv2D)	(None, 384, 256, 64)	8,256	up_sampling2d_5[0][0]
add_5 (Add)	(None, 384, 256, 64)	0	up_sampling2d_4[0][0]
activation_13 (Activation)	(None, 384, 256, 64)	0	add_5[0][0]
conv2d_transpose_6 (Conv2DTranspose)	(None, 384, 256, 32)	18,464	activation_13[0][0]
batch_normalization (BatchNormalization)	(None, 384, 256, 32)	128	conv2d_transpose_6[0][0]
activation_14 (Activation)	(None, 384, 256, 32)	0	batch_normalization[0][0]
conv2d_transpose_7 (Conv2DTranspose)	(None, 384, 256, 32)	9,248	activation_14[0][0]
batch_normalization (BatchNormalization)	(None, 384, 256, 32)	128	conv2d_transpose_7[0][0]
up_sampling2d_7 (UpSampling2D)	(None, 768, 512, 64)	0	add_5[0][0]
up_sampling2d_6 (UpSampling2D)	(None, 768, 512, 32)	0	batch_normalization[0][0]
conv2d_7 (Conv2D)	(None, 768, 512, 32)	2,880	up_sampling2d_7[0][0]
add_6 (Add)	(None, 768, 512, 32)	0	up_sampling2d_6[0][0]
conv2d_8 (Conv2D)	(None, 768, 512, 1)	289	add_6[0][0]

Figure 28: Architecture du U-Net

Initialement, nous avons testé ce modèle dans les mêmes conditions que le CNN, sans appliquer de prétraitements particuliers aux images. Bien que les résultats aient été légèrement plus lisibles que ceux obtenus avec le CNN, la segmentation n'était toujours pas satisfaisante du tout.

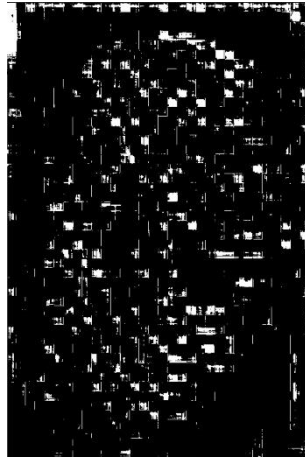


Figure 29: Résultats sans prétraitements

La dernière approche que nous avons testée impliquait le même modèle U-Net, mais cette fois avec des images ayant subi des opérations d'ouverture et de fermeture. Nous avons également augmenté notre jeu de données en appliquant une série de transformations simples aux images, telles que des rotations et des filtrages gaussiens, ...

Cette méthode a conduit à des améliorations notables des résultats. Bien qu'il ne nous semble plus utile de poursuivre d'autres tests dans l'immédiat, nous pensons qu'avec un accès à davantage de puissance de calcul, nous pourrions significativement améliorer les performances de nos modèles.

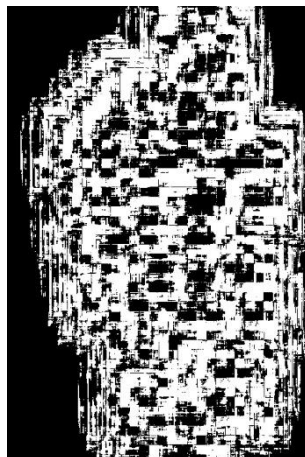


Figure 30: Résultat avec prétraitements

VIII. Conclusion

La segmentation précise de la lame criblée reste un défi en raison de sa structure complexe et de l'importance de ses fonctions. Nos résultats montrent que certaines méthodes, telles que l'augmentation du contraste par manipulation manuelle et l'application du filtre top hat, offrent des performances relativement bonnes en termes de précision et de F1 score notamment sur la LC1. En revanche, selon l'image rencontrée, les résultats ne seront pas forcément ceux espérer.

Il ne faut pas oublier que chaque situation est différente et demande une solution adaptée, donc selon le cas, une méthode peut être préférable à l'autre pour maximiser plutôt la précision ou plutôt le rappel.

De nombreuses additions pourraient être apportée à nos transformation afin d'améliorer l'ensemble de ces résultats, mais nous envisageons sinon d'explorer des techniques avancées de machine learning et de deep learning, ainsi que des collaborations pour obtenir un ensemble de données plus large. En effet même si nos méthodes combinées offrent une amélioration notable, mais le manque de données pour l'entraînement des modèles d'IA reste un obstacle majeur. Pour l'avenir, nous envisageons d'explorer des collaborations pour accéder à un plus grand ensemble de données et d'améliorer notre modèle de deep learning avec des techniques plus robustes de transfer learning et d'augmentation de données.

Notre code se trouve sur un git à l'adresse : <https://github.com/Lu6m/Optic-Nerve-Analysis>