



INTRODUCTION AU TRAITEMENT DES IMAGES

Rapport TP03 IMA201

Réalisé par:

Adnane EL BOUHALI

Encadré par:

GOUSSEAU Yann, LECLAIRE Arthur, LESNÉ Gwilherm, ROUX
Michel

Etablissement :

Télécom Paris

Filière :

Image

Année universitaire:

2023 - 2024

Table des matières

INTRODUCTION AU TRAITEMENT DES IMAGES.....	1
Table des matières.....	2
1 Détection de contours.....	3
1.1 Filtre de gradient local par masque.....	3
1.2 Maximum du gradient filtre dans la direction du gradient.....	4
1.3 Filtre récursif de Deriche.....	6
1.4 Passage par zéro du laplacien.....	8
1.5 Changez d'image.....	10
2 Seuillage avec hystérésis.....	10
3 Segmentation par classification : K-moyennes.....	13
3.1 Image a niveaux de gris.....	13
3.2 Image en couleur.....	14

1 Détection de contours

1.1 Filtre de gradient local par masque

Rappelez l'intérêt du filtre de Sobel, par rapport au filtre différence, qui calcule une dérivée par la simple différence entre deux pixels voisins.

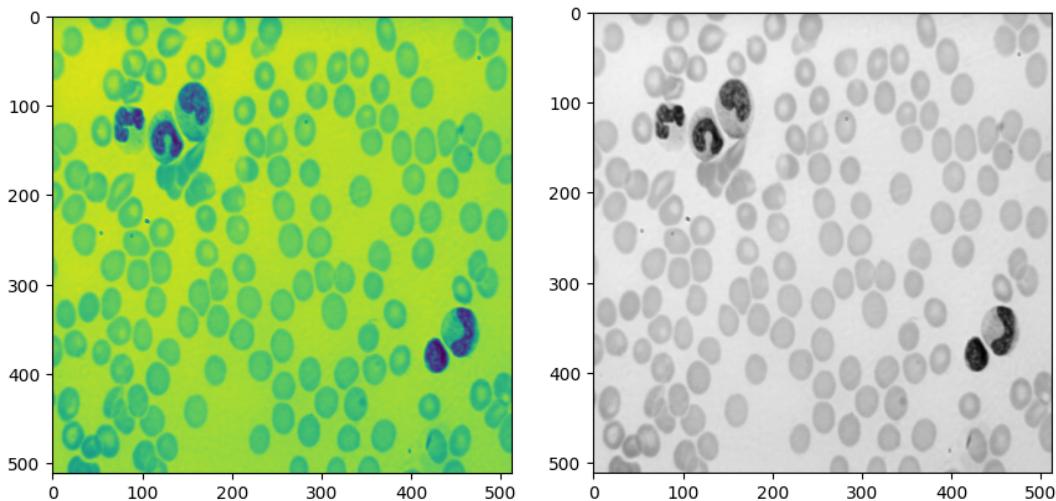
Le filtre de Sobel est préféré au filtre de différence simple pour la détection de contours car il réduit le bruit, détecte les contours de manière plus efficace et fournit des informations sur leur direction.

Est-il nécessaire de faire un filtre passe-bas de l'image avant d'utiliser le filtre de Sobel ?

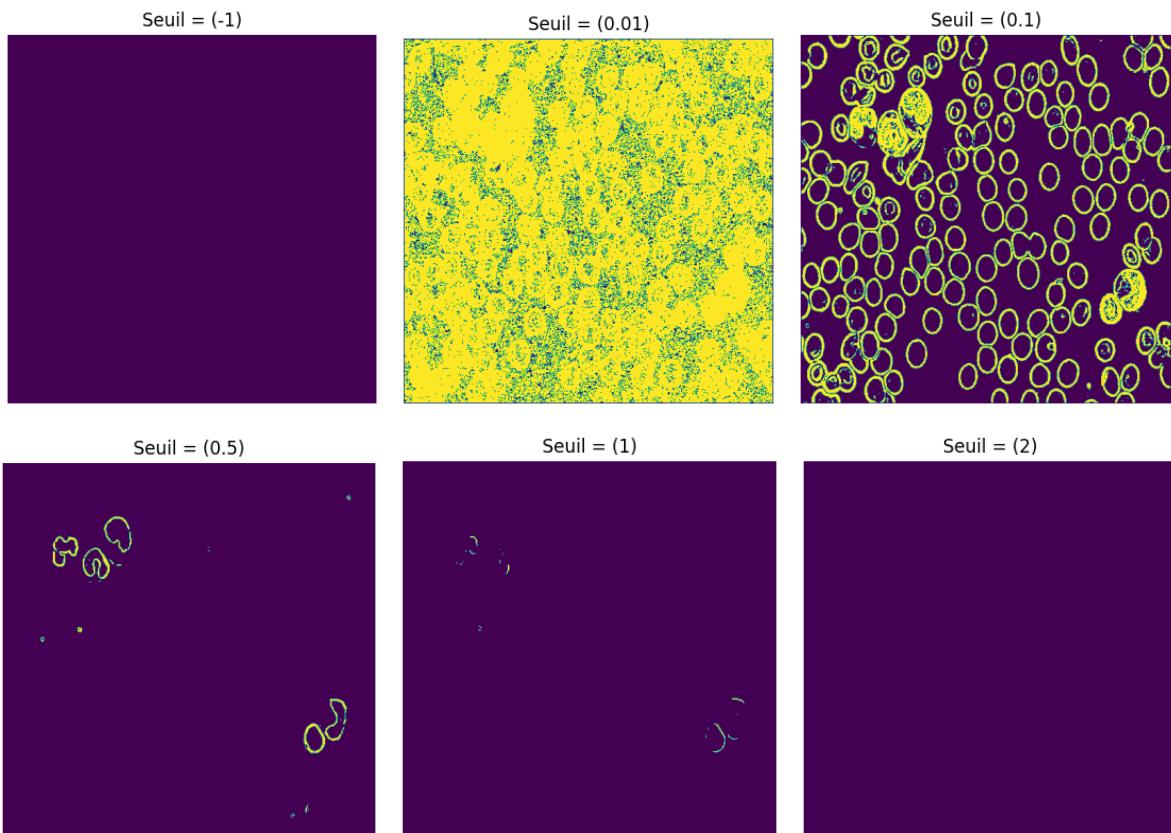
Non, car le filtre de Sobel réduit déjà le bruit, donc pas la peine de faire passer un filtre passe bas.

Commentez la qualité des contours obtenus (robustesse au bruit, continuité, épaisseur, position...) quand l'on fait varier ce seuil.

D'abord, voici l'image originale, et l'image en niveaux de gris.



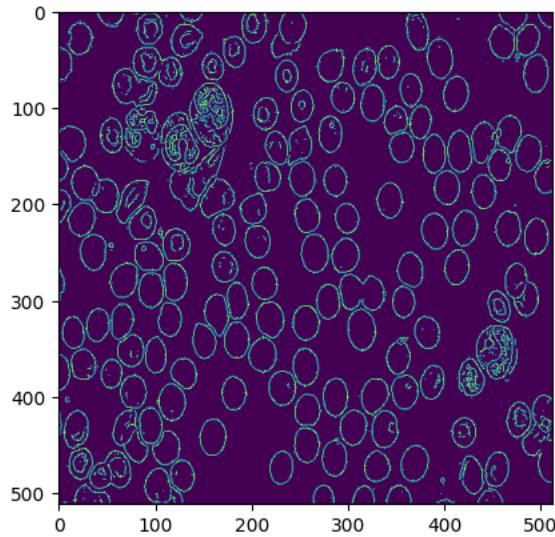
Et voici les contours obtenus en faisant varier le seuil.



Selon les critères de robustesse au bruit, continuité, épaisseur et position, la valeur seuil = 0.1 donne les meilleurs résultats comme on peut le voir sur les figures ci-dessus.

1.2 Maximum du gradient filtre dans la direction du gradient

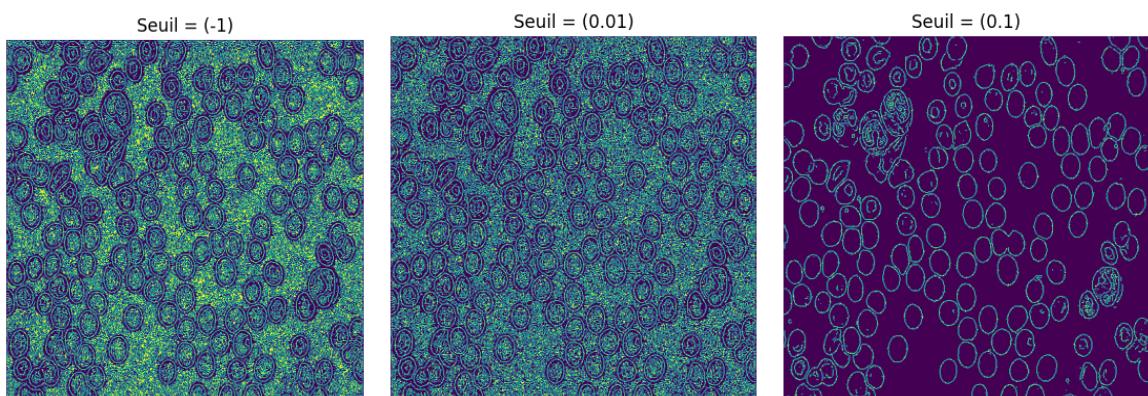
Quel critère de qualité est optimisé par ce procédé ?

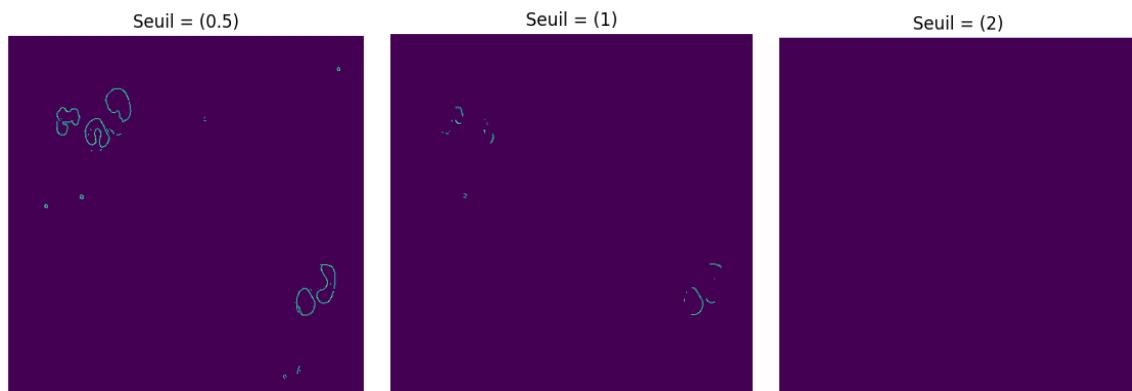


Comme on peut le voir dans l'image ci-dessus, le critère de qualité optimisé est la finesse des contours. En effet, on a des contours qui correspondent à exactement un pixel.

Il est possible d'éliminer les contours dont la norme est inférieur à un seuil donne. Commentez les résultats obtenus en terme de position et de continuité des contours, et de robustesse au bruit en faisant varier ce seuil.

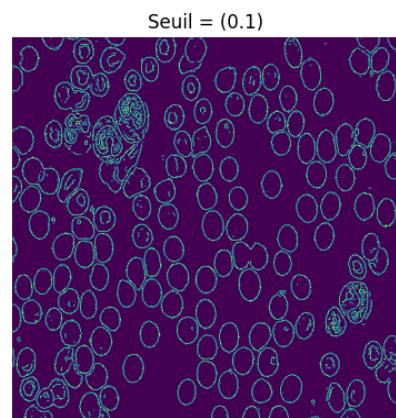
Encore une fois, selon les critères de robustesse au bruit, continuité, épaisseur et position, la valeur seuil = 0.1 donne les meilleurs résultats comme on peut le voir sur les figures ci-dessus.





Cherchez à fixer le seuil sur la norme de façon à obtenir un compromis entre robustesse au bruit et continuité des contours.

Le seuil 0.1 permet d'obtenir un compromis entre robustesse au bruit et continuité des contours.



1.3 Filtre récursif de Deriche

**Dans le fichier, mrlab.py, des erreurs ont été commises dans les fonctions de-
dericheGradX et dericheGradY. A vous de corriger ces fonctions (uniquement
au niveau des lignes indiquées) an de mettre en uvre la récursivité.**

```

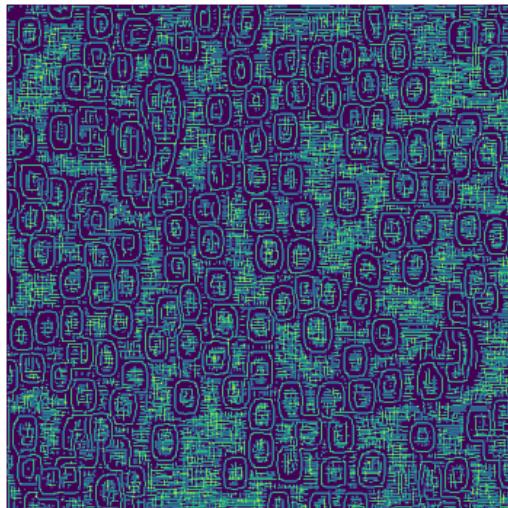
44 def dericheGradX(ima,alpha):
45
46
47     nl,nc=ima.shape
48     ae=math.exp(-alpha)
49     c=-(1-ae)*(1-ae)/ae
50
51     b1=np.zeros(nc)
52     b2=np.zeros(nc)
53
54     gradx=np.zeros((nl,nc))
55
56
57     #gradx=np.zeros(nl,nc)
58     for i in range(nl):
59
60         l=ima[i,:].copy()
61
62         for j in range(2,nc):
63             b1[j] = l[j-1] + 2*ae*b1[j-1] - ae*ae*b1[j-2] # LIGNE A MODIFIER
64             b1[0]=b1[2]
65             b1[1]=b1[2]
66
67             for i in range(nc-3,-1,-1):
68                 b2[j] = l[j+1] + 2*ae*b2[j+1] - ae*ae*b2[j+2] # LIGNE A MODIFIER
69                 b2[nc-2]=b2[nc-3]
70                 b2[nc-1]=b2[nc-3]
71
72             gradx[i,:]=c*ae*(b1-b2)
73
74     return gradx

```

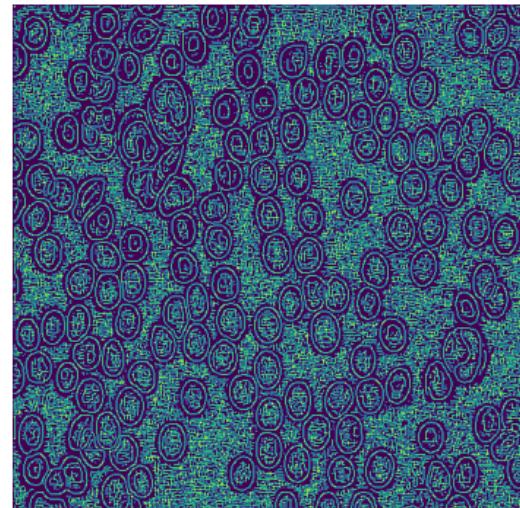
On fait de même pour la fonction `dericheGradY`.

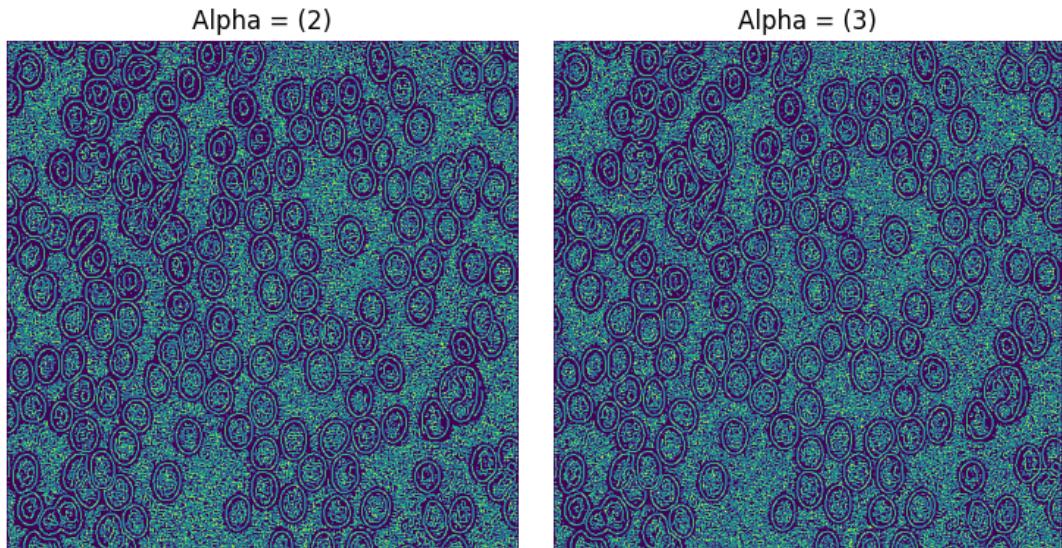
Testez la détection de contours avec ce filtre sur plusieurs images. Décrivez l'effet du paramètre alpha sur les résultats de la segmentation (faites varier ce paramètre sur l'intervalle 0,3...3,0).

Alpha = (0.3)



Alpha = (1)





On remarque que plus le paramètre alpha augmente, plus la finesse des contours augmente.

Le temps de calcul dépend-il de la valeur de alpha? Expliquez pourquoi.

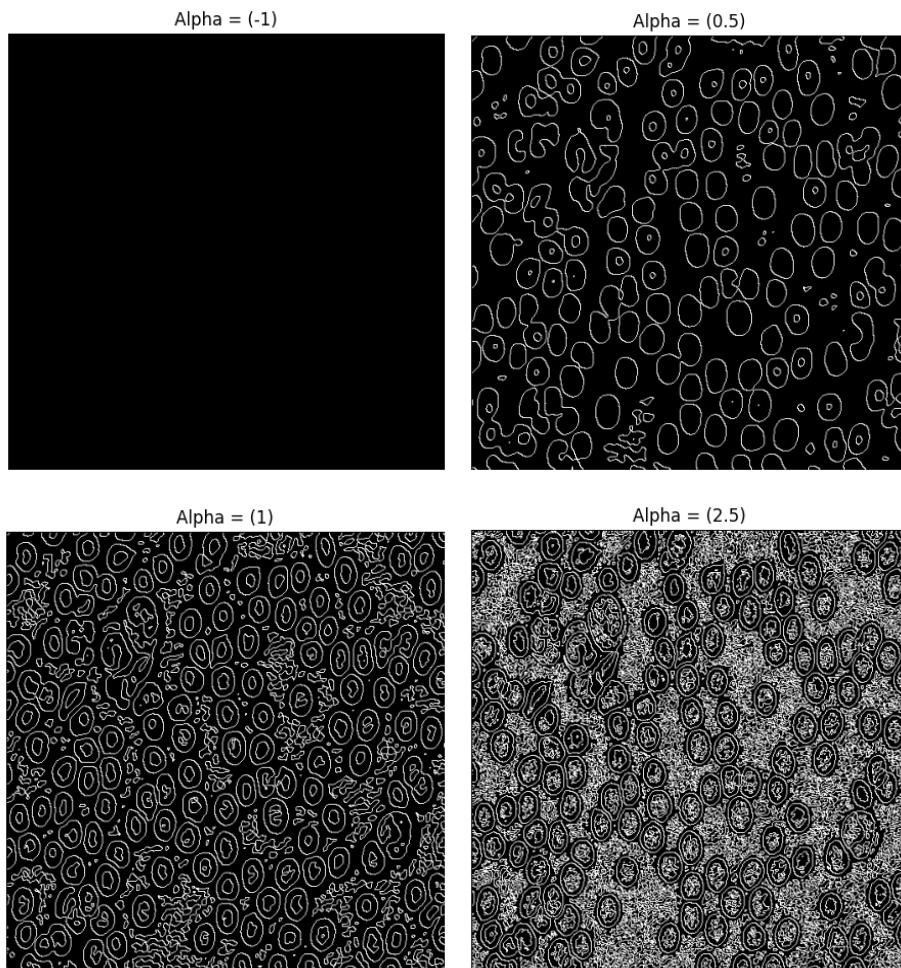
Non, la valeur de alpha n'influe pas sur le temps de calcul, car il n'est utilisé que pour calculer la constante c, et l'exponentiel moins alpha, ce qui est constant.

Comment et dans quel but les fonctions `dericheSmoothX` et `dericheSmoothY` sont-elles utilisées (cf. le filtre de Sobel).

Les fonctions **dericheSmoothX** et **dericheSmoothY** sont utilisées pour effectuer une opération de filtrage préliminaire sur l'image. Après, les gradients en X et Y sont calculés à l'aide de **dericheGradX** et **dericheGradY**. Ces gradients sont ensuite utilisés pour calculer la norme des gradients (**norme**) et la direction du gradient (**direction**), avant d'appliquer le filtre afin de détecter les contours.

1.4 Passage par zéro du laplacien

Quel est l'effet du paramètre alpha sur les résultats?

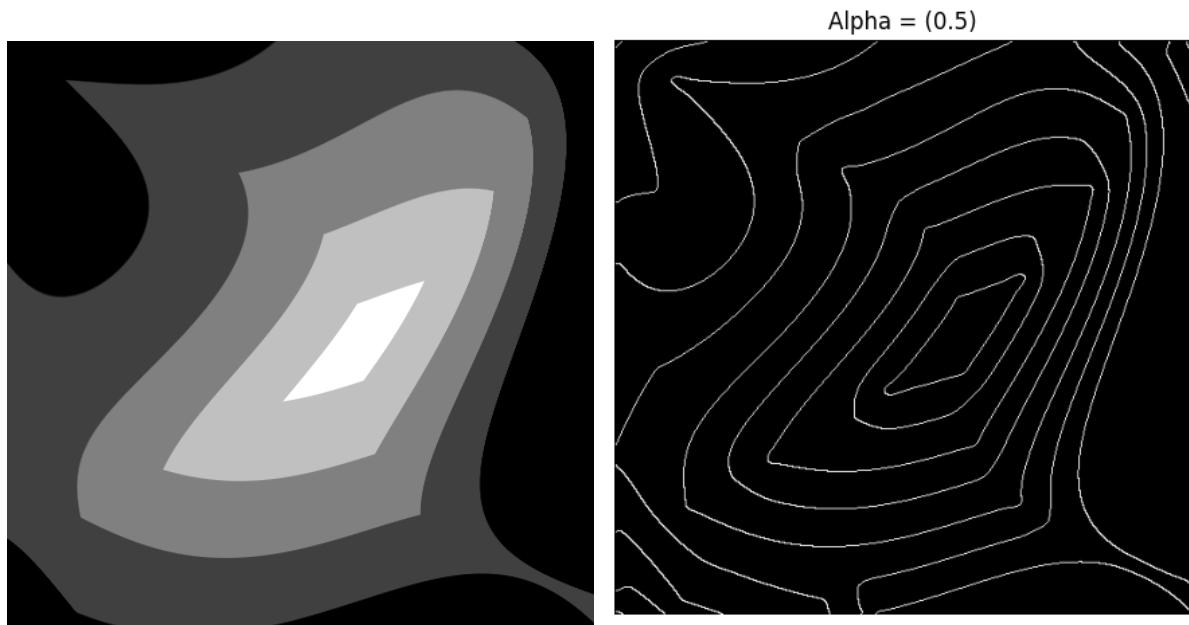


Lorsque nous augmentons le paramètre alpha, l'intensité du bruit détecté augmente. Tandis que l'intensité du bruit diminue lorsque nous le réduisons.

Sur l'image cell.tif, quelles sont les principales différences par rapport aux résultats fournis par les opérateurs vus précédemment (contours, Deriche) ?

La différence principale réside dans le fait que le laplacien génère des contours fermés, contrairement aux méthodes précédentes qui peuvent produire des lignes ouvertes.

Sur l'image pyramide.tif, comment est-il possible de supprimer les faux contours créés par cette approche ?



Pour éliminer les contours indésirables créés par la méthode, on peut appliquer un filtre pour réduire le bruit d'abord. Cela améliorera la précision en préservant les contours importants.

1.5 Changez d'image

Quel opérateur choisiriez-vous pour segmenter l'image pyra-gauss.tif ?

L'image pyra-gauss.tif présente du bruit, ce qui rend le laplacien inadapté car il réagit fortement aux petites variations de gradient. Dans cette situation, la meilleure solution serait d'opter pour un filtre de Deriche.

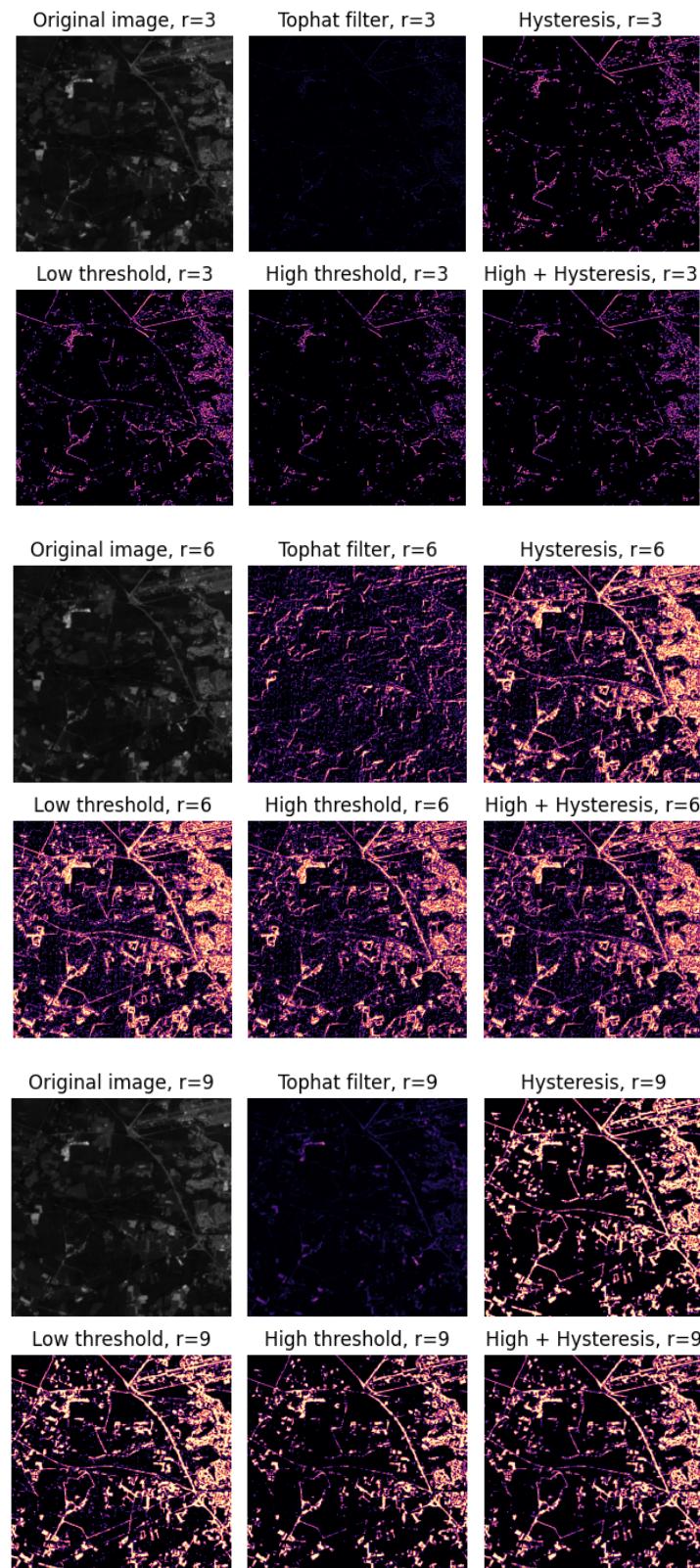
Quels seraient les pré-traitements et les post-traitements à effectuer ?

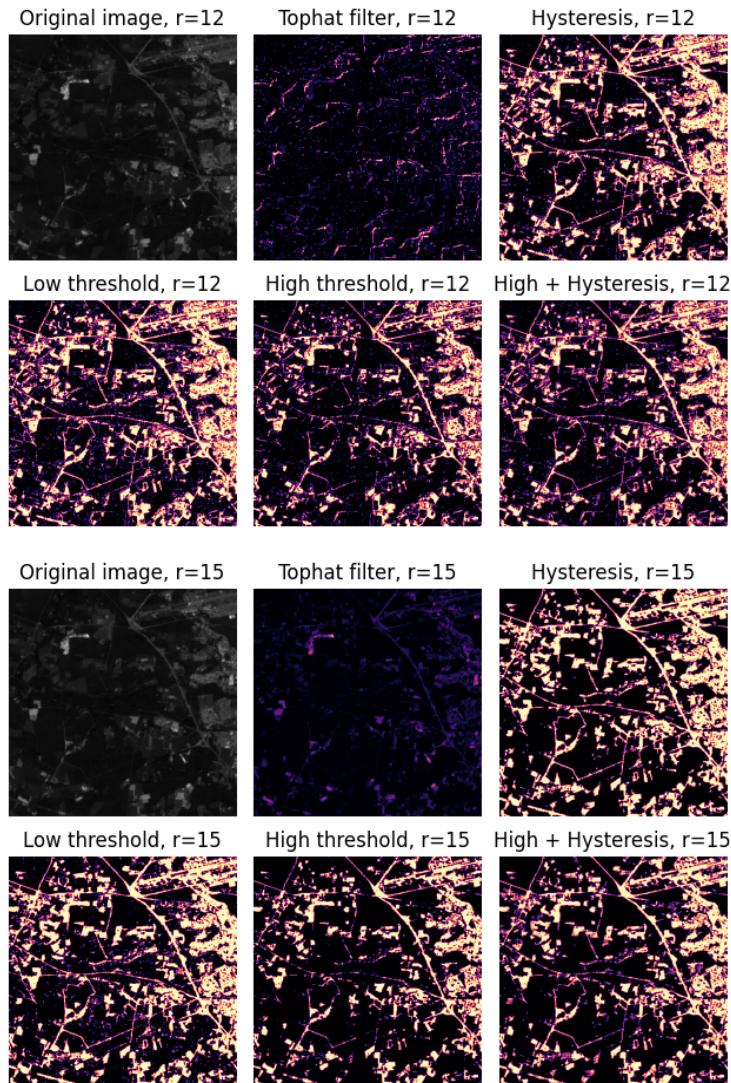
En ce qui concerne les étapes de pré-traitements, il est recommandé d'appliquer un filtre gaussien, parmi d'autres options. Pour les étapes de post-traitements, on peut utiliser des techniques telles que la détection du maximum local du gradient, le seuillage, l'hystérésis et la fermeture.

2 Seuillage avec hystérésis

Appliquez le filtre du Chapeau haut de forme (tophat) à une image SPOT pour effectuer une détection de lignes.

Modifiez le rayon de l'élément structurant utilisé pour calculer le ltre tophat, et indiquez comment évoluent les lignes détectées.

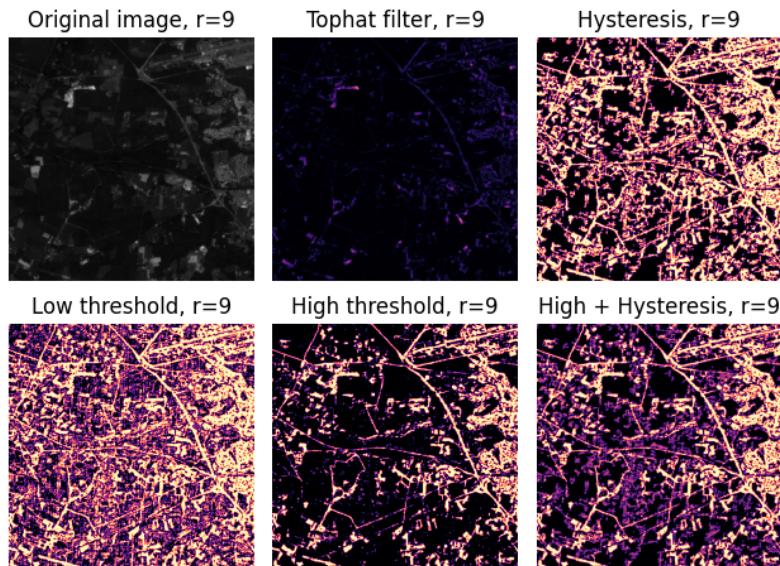




On remarque que plus le rayon augmente, plus les contours deviennent plus marqués.

Modifiez les valeurs des deux seuils, et examinez comment les lignes sont supprimées ou préservées. Quels sont les seuils qui donnent, à votre avis, le meilleur résultat ?

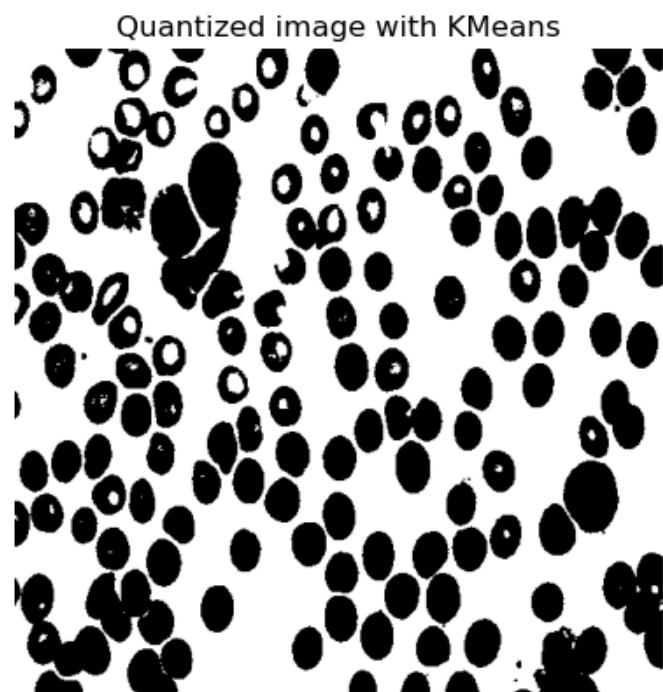
Après avoir testé plusieurs combinaisons, je trouve que le meilleur résultat est : low = 1, high = 3, rayon = 9.



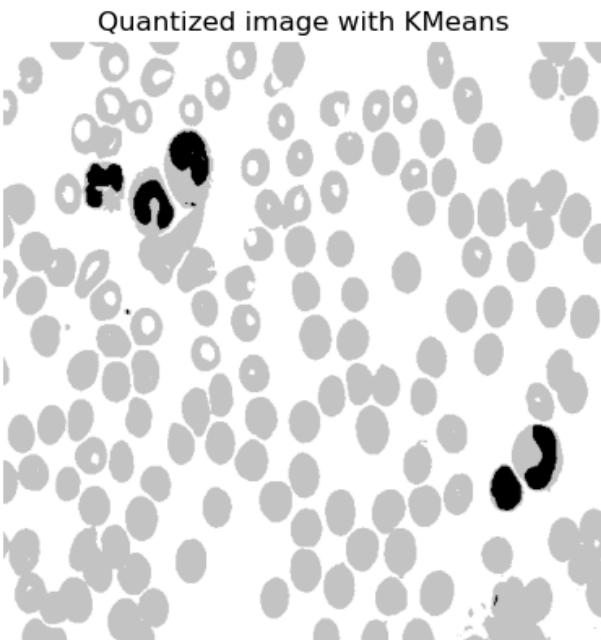
3 Segmentation par classification : K-moyennes

3.1 Image à niveaux de gris

Testez l'algorithme des k-moyennes sur l'image cell.tif pour une classification en 2 classes. Cette classification segmente-t-elle correctement les différents types de cellules ? Si non, que proposez-vous ?



On ne peut pas distinguer les cellules avec seulement deux classes. On voit seulement où il y a des cellules et où il n'y en a pas. C'est pourquoi tout est noir pour les cellules et blanc pour le fond. Pour résoudre cela, ajoutons une troisième classe.



Testez les différentes possibilités pour initialiser les classes. Décrivez si possible ces différentes méthodes.

Qu'il y ait trois classes, que les centres soient positionnés de façon déterministe ou aléatoire, le résultat reste le même. Ajouter plus de classes n'affecte pas la distinction des cellules différentes.

3.2 Image en couleur

Testez l'algorithme sur l'image `fleur.tif` pour une classification en 10 classes, les centres des classes initiaux étant tirés aléatoirement).

Original image



Quantized image with K-Means: 10 colours

