IMA201
Travaux Pratique 3: Segmentation des images

Alice Valença De Lorenci

October 10, 2022

1 Détection de contours

1.1 Filtre de gradient local par masque

The local gradient filters, such as the difference, Roberts, Prewitt and Sobel filters, are used to detect edges in the following way: the filter is convolved with the image to calculate approximations of the derivative in one direction, then the filter is rotated by $\pi/2$ and the derivative in calculated in the other direction. Both measures are then combined to obtain the norm of the gradient:

$$G = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \tag{1}$$

The direction of the gradient can also be computed:

$$\Phi = \arctan \left[\frac{\partial f}{\partial x} / \frac{\partial f}{\partial y} \right] \tag{2}$$

Those steps are followed by post-treatment methods such as thresholding, detection of maximums of the gradient on the direction of the gradient, thresholding by hysteresis or tracking and closing of contours.

Rappelez l'intérêt du filtre de Sobel, par rapport au filtre différence, qui calcule une dérivée par la simple différence entre deux pixels voisins.
The simple difference filter (2x2), that computes the difference between two neighboring pixels is centered between pixels, which causes a shifting of the image. Meanwhile, by introducing a column (or row) of zeros between the asymmetric coefficients, the Sobel filter (3x3) is centered on the pixel itself. Furthermore, the Sobel filter computes the gradient in one direction and a weighted mean in the other, making it more robust to noise.

Est-il nécessaire de faire un filtre passe-bas de l'image avant d'utiliser le filtre de Sobel?
The interest of using a low pass filter is to denoise the image, improving the quality of the edges, which, by extension, improves the quality of edge detection. Therefore, the need to use a low pass filter before applying the Sobel filter depends on the level of noise of the image, for very low levels the robustness of the Sobel filter might be enough.

Le seuillage de la norme du gradient permet d'obtenir des contours. Commentez la qualité des contours obtenus (robustesse au bruit, continuité, épaisseur, position...) quand l'on fait varier ce seuil.

When we set a very low threshold, the contours detected are continuous, however they are very susceptible to noise and quite thick, which implies uncertainty on their true location, also we observe the detection of false contours. By increasing the threshold, the contours become thinner (better estimate of position) and more robust to noise, at the expense of continuity. Also, although there is a reduction in the amount of false contours detected, part of the true ones are also lost.

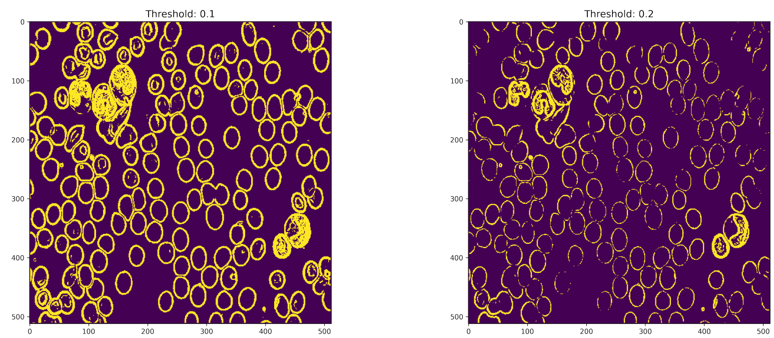


Figure 1: Thresholding of the norm of the gradient.

1.2 Maximum du gradient filtré dans la direction du gradient

Quel critère de qualité est optimisé par ce procédé?
The Canny criteria analytically define desirable characteristics for contour detection:

- good detection, as many edges as possible should be accurately detected, meaning that the detector should have a strong response even to weak contours and should maximize the signal to noise ratio;
- good localization, the points detected should be at the center of contour;
- unique response, each contour should be detected only once and noise should not create false edges.

The procedure of detecting the maximum of gradient on the direction of the gradient isn't robust to noise and detects many false edges, therefore it doesn't optimize the first and third criteria. However, by detecting the local maximums of the gradient, this procedure finds the accurate location of the contours and yields thin lines, optimizing the second criterion.

Il est possible d'éliminer les contours dont la norme est inférieur à un seuil donné. Commentez les résultats obtenus en terme de position et de continuité des contours, et de robustesse au bruit en faisant varier ce seuil.
When we apply the procedure of detecting the maximum of the gradient on the direction of the gradient, we must filter the contours detected by thresholding according to the norm of the gradient, in order to eliminate the false contours.

The higher the threshold, the less false contours we detect (more robust to noise), however we lose continuity and part of the true contours. The localization and thickness of the contours isn't affected.

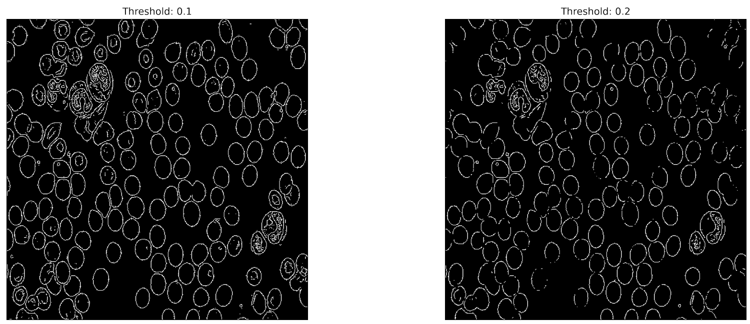
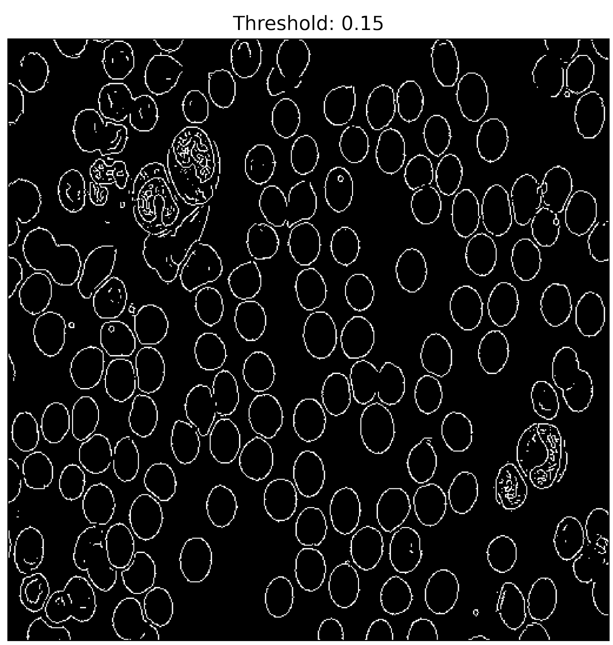


Figure 2: Maximum of the gradient in the gradient direction combined with thresholding of the norm of the gradient.

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

Varying the threshold used to discard contours based on the norm of the gradient, a good compromise between robustness to noise and continuity of the contours seems to be found for a threshold of 0.15: most of the spurious contours are eliminated, although some micro structures remain, and the majority of the cells are detected.



1.3 Filtre récursif de Deriche

In the `deriche.py` python file, Deriche's filter is used to approximate the gradient of the image, then the contours are detected using the maximum of the gradient in the gradient direction procedure, followed by gradient magnitude thresholding.

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

```python
def dericheGradX(ima,alpha):


    nl,nc=ima.shape
    ae=math.exp(-alpha)
    c=-(1-ae)*(1-ae)/ae

    b1=np.zeros(nc)
    b2=np.zeros(nc)

    gradx=np.zeros((nl,nc))

    for i in range(nl):

        l=ima[i,:].copy()

        for j in range(2,nc):
            b1[j] = l[j-1] + 2*ae*b1[j-1] - ae*ae*b1[j-2]
        b1[0]=b1[2]
        b1[1]=b1[2]

        for j in range(nc-3,-1,-1):
            b2[j] = l[j+1] + 2*ae*b2[j+1] - ae*ae*b2[j+2]
        b2[nc-2]=b2[nc-3]
        b2[nc-1]=b2[nc-3]

        gradx[i,:]=c*ae*(b1-b2);

    return gradx

def dericheGradY(ima,alpha):


    nl,nc=ima.shape
    ae=math.exp(-alpha)
    c=-(1-ae)*(1-ae)/ae

    b1=np.zeros(nl)
    b2=np.zeros(nl)

    grady=np.zeros((nl,nc))

    for i in range(nc):

        l=ima[:,i].copy()
```

```

        for j in range(2,nl):
            b1[j] = l[j-1] + 2*ae*b1[j-1] - ae*ae*b1[j-2]
        b1[0]=b1[2]
        b1[1]=b1[2]

        for j in range(nl-3,-1,-1):
            b2[j] = l[j+1] + 2*ae*b2[j+1] - ae*ae*b2[j+2]
        b2[nl-1]=b2[nl-3]
        b2[nl-2]=b2[nl-3]

        grady[:,i]=c*ae*(b1-b2);


    return grady
```

Testez la détection de contours avec ce filtre sur plusieurs images. Décrivez l'effet du paramètre α sur les résultats de la segmentation (faites varier ce paramètre sur l'intervalle $0.3, ..., 3.0$**).**

The α scale parameter of Deriche's filter corresponds to the inverse of the σ parameter of Canny's filter, which indicates bellow which distance two parallel contours are merged. Therefore, the α parameter can be adjusted to filter out noise by linking adjacent edges into long, smooth, continuous contours.

By reducing the α parameter, we observe the reduction of noise and spurious contour detection, however, by making it too small, the edges of different objects start to merge into a single contour.

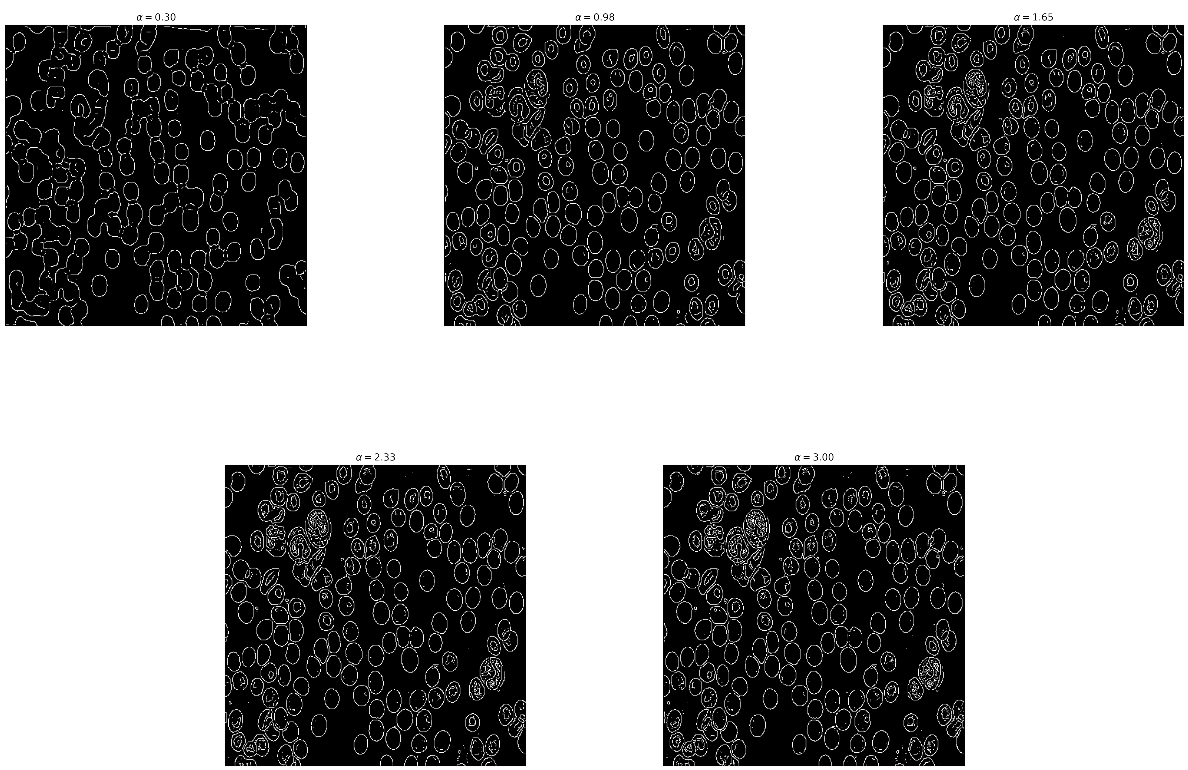


Figure 3: Deriche filter.

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

The execution time of Deriche's edge detector does not depend on the value of α, since it's only used to calculate the coefficients of the recursive expression, once the coefficients are calculated its values can be stored. The value of the parameter does not change the number of operations.

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

The Deriche filter is initially defined in one dimension, it is then extend to two dimensions by the cross application of two filters, one in the x direction (detection of the vertical component of edges), and one in the y direction (detection of the horizontal component of edges). Furthermore, in the direction of the contour a smoothing function is defined, which allows the filtering of noise, this function corresponds to the `dericheSmoothX` and `dericheSmoothY` procedures.

In conclusion, an edge detector filter is composed of two derivative estimators, one in the x and the other in the y direction. Each of these detectors is composed of the product of two function, taking for instance the detector in the x direction, we have a low pass filter along Oy (smoothing function) and a high pass filter along Ox.

1.4 Passage par zéro du laplacien

In the `laplacien.py` python file, successive applications of Deriche's filter are used to approximate the Laplacian of the image, then the contours are detected as the zero crossings of the Laplacian. This method is justified in the absence of strong curvatures, in which case the Laplacian is a good approximation of the second derivative of the image in the direction of the gradient, thus the zero crossing of the Laplacian corresponds to the maximum of the gradient in the gradient direction.

This method has the advantage that the zeros of the Laplacian constitute a collection of closed lines.

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

By reducing the α parameter, we observe the reduction of noise and spurious contour detection, however, by making it too small, the edges of different objects start to merge into a single contour.

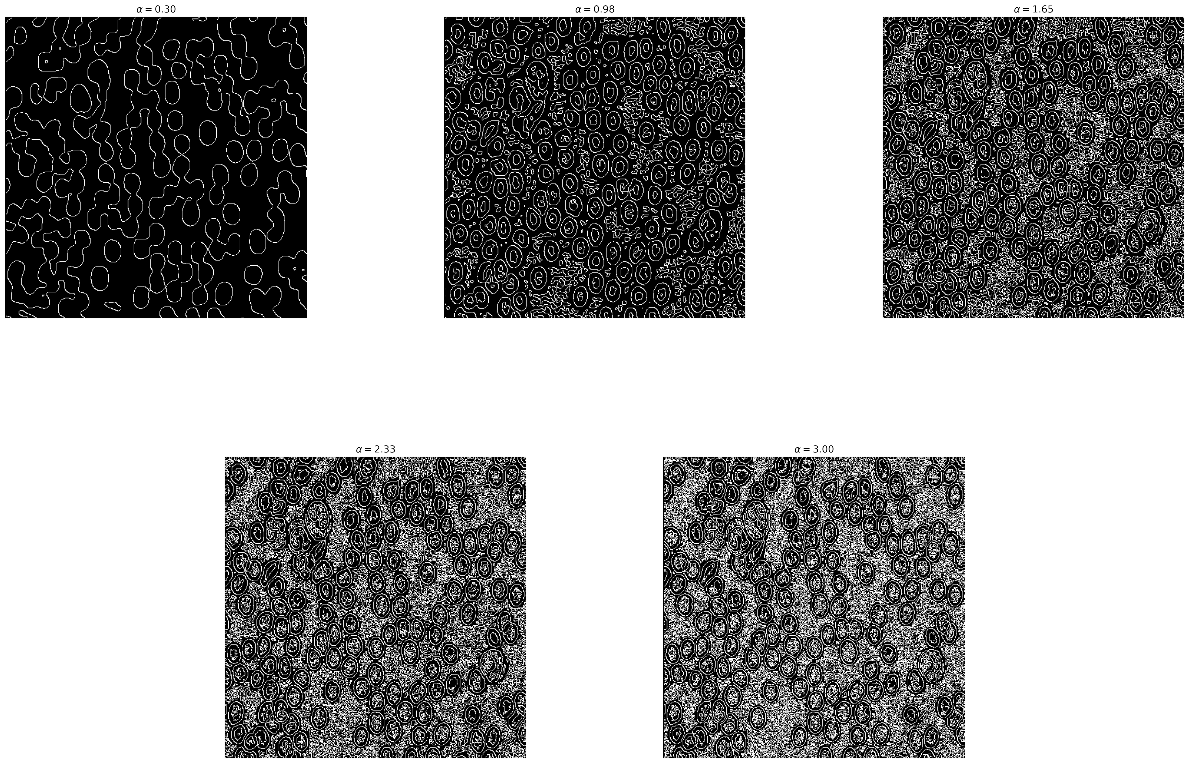


Figure 4: Zero crossings of the Laplacian.

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

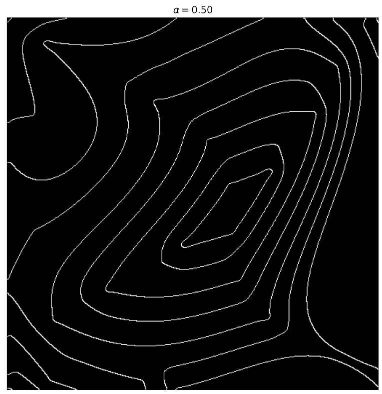
The local gradient filters, such as the Sobel filter, generate only local edge data, instead of recovering the global structure of a boundary and are highly sensitive to noise. The procedure of maximizing the gradient in the gradient direction is also oblivious to global structures, although not as much as local masks, and is sensitive to noise, but at least optimizes the localization criterion. On the other hand, Deriche's filter and the zero crossings of the Laplacian procedure are able to abstract global structures, through adjustment of the α parameter, also, Deriche's filter in particular optimizes all of Canny's criteria. The Laplacian approach has the advantage of detecting closed contours.

Comparing the results obtained on the `cell.tif` image for the Laplacian method and Deriche's filter, we observe that for higher values of α the first is less robust to noise. This is explained by the fact that the Laplacian method does not take into account the norm of the gradient.

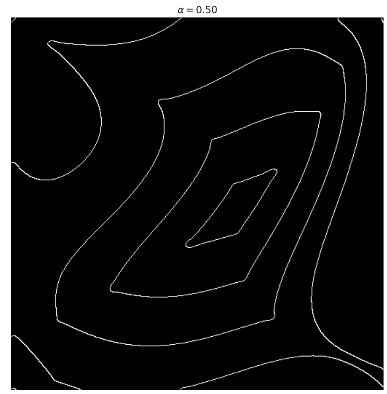
Sur l'image `pyramide.tif`, comment est-il possible de supprimer les faux con-tours créés par cette approche?

The Laplacian approach has the disadvantage that the Laplacian of inflection points of a function is also equal to zero, resulting in the detection of false contours on step-like images, such as `pyramide.tif`. To eliminate such contours, it is possible to eliminate contours whose norm is inferior to a given threshold because as they do not correspond to actual edges, the norm of the gradient will be very small in such regions.

(a) Original image.



(b) Zero crossings of the Laplacian.



(c) Zero crossings of the Laplacian with norm thresholding.

1.5 Changez d'image

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

Since the image `pyra-gauss.tif` is noisy and step-like, I would choose Deriche's operator as it is the most robust to noise and optimizes Canny's criteria, also the detection of false contours dua to noise can be controlled through adjustment of the α parameter.

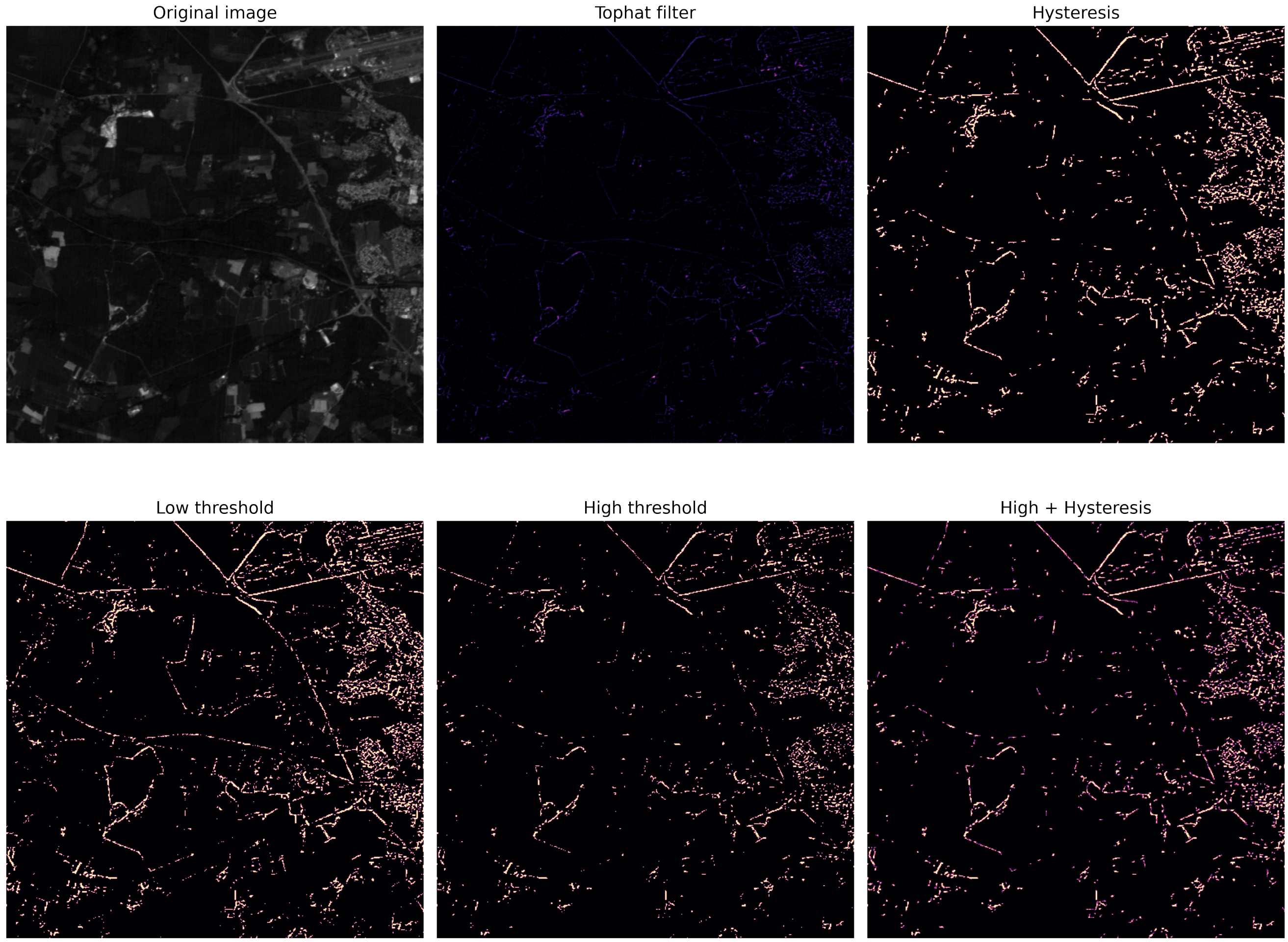
Quels seraient les pré-traitements et les post-traitements à effectuer?

Since the image `pyra-gauss.tif` is noisy, the pre processing should consist of a denoising step (concolution with a low pass filter). And the post processing would include the detection of the maximum of the gradient in the gradient direction combined with contour suppression based on gradient norm thresholding.

2 Seuillage avec hystérésis

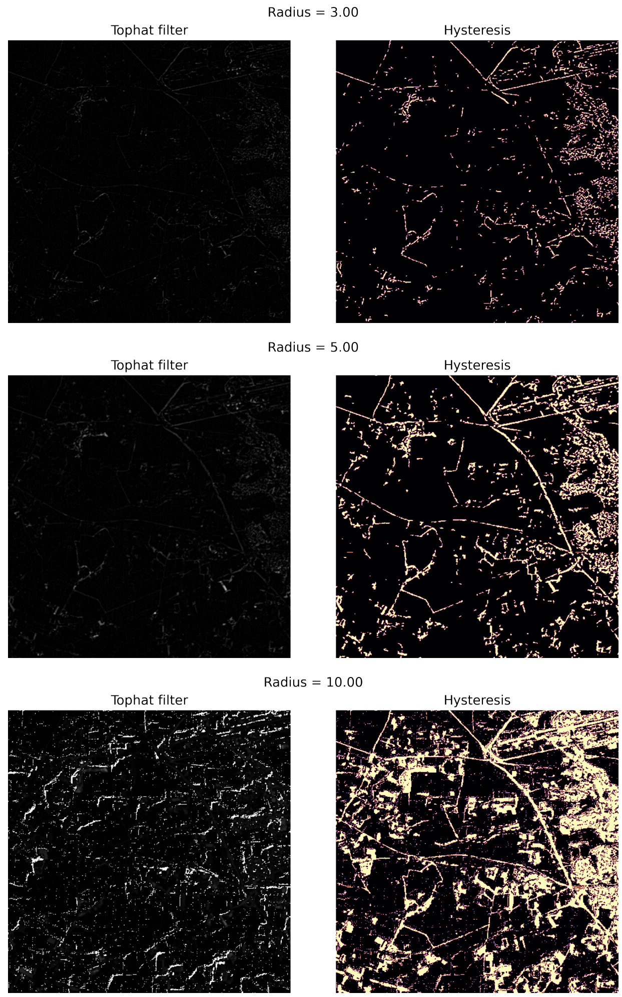
2.1 Application à la détection de lignes

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



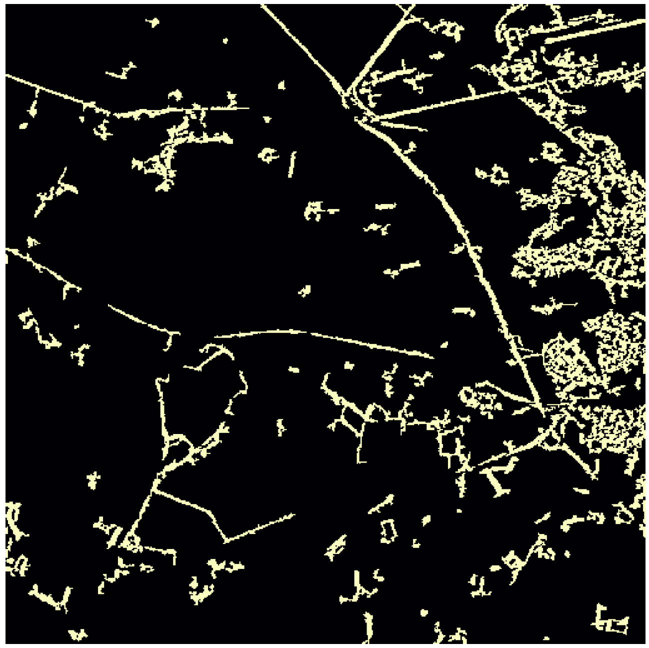
Modifiez le rayon de l'élément structurant utilisé pour calculer le filtre `tophat`, et indiquez comment évoluent les lignes détectées.

By increasing the radius the detected lines become thicker and more continuous, but there is also an increase in the detection of false contours.



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?

In my opinion, for a radius of 5, the high and low thresholds that give the best results for `spot.tif` are 2 and 15, respectively. Those were the values that yielded the best compromise between detecting the roads on the image and reducing the amount of spurious contours.



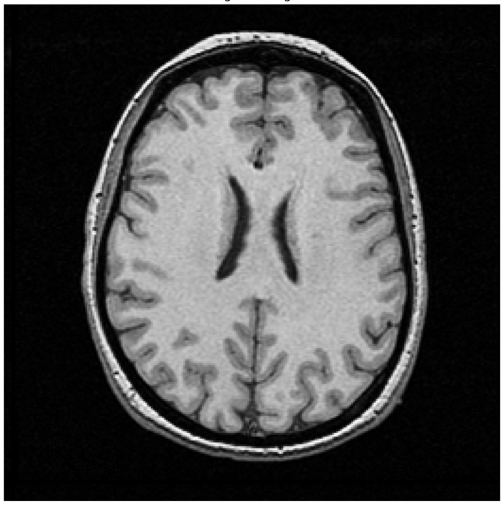
Appliquez le seuillage par hystérésis pour améliorer la détection de contours obtenue avec un des opérateurs vus précédemment sur une image de votre choix. Précisez la mise en oeuvre que vous proposez et commentez les résultats.

Hysteresis thresholding will be applied to improve contour detection on the image `cell.tif` using the Sobel operator combined with the maximum of the gradient in the gradient direction procedure (Section 1.2). First of all, as a pre processing step, a low pass Gaussian filter is used to denoise the image. The Sobel operator is then applied to the image to compute the gradient (norm and direction) and the contours are detected as the maximum of the gradient in the gradient direction.

Next, hysteresis thresholding is used to improve the quality of the detection: a threshold is chosen sufficiently low so that the lines detected when thresholding the gradient norm are continuous, this threshold is highly sensitive to noise; a second threshold is chosen high enough to preserve only points that belong to valid contours. The second image is then used to select the contours of the first image that contain at least one point kept by the high threshold. The resulting image is used as a mask that is applied to the contours detected by the maximum of the gradient in the gradient direction procedure.

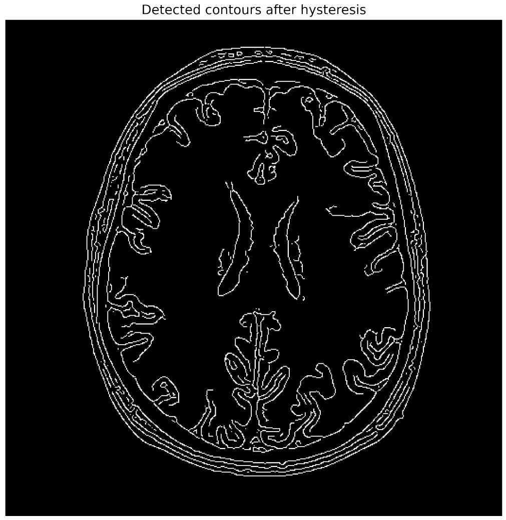
The maximum of the gradient in the gradient direction procedure guarantees a good localization, while hysteresis thresholding allows the suppression of spurious contours without sacrificing continuity.

Original image



Maximum of the gradient in the gradient direction

Detected contours after hysteresis

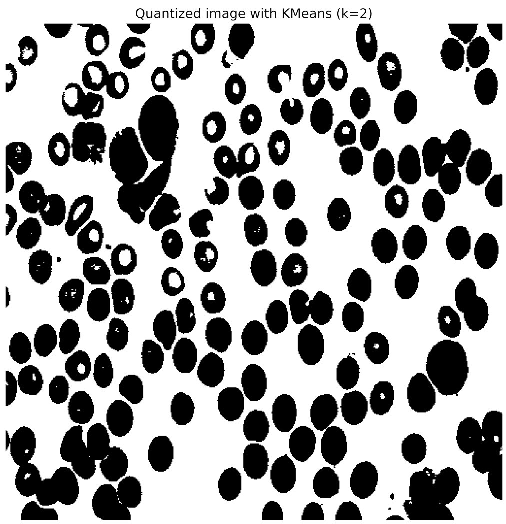


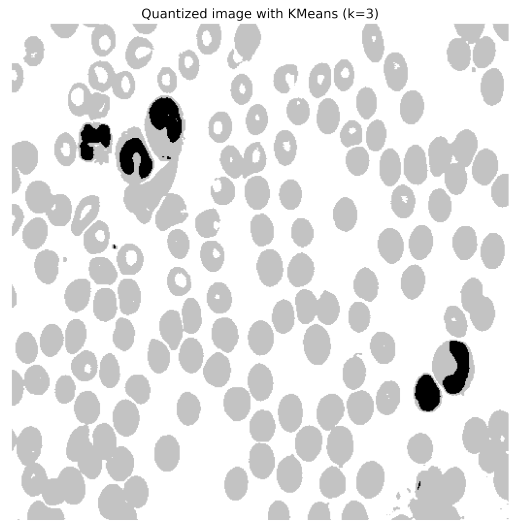
3 Segmentation par classification : K-moyennes

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?

The k-means algorithm, for $k = 2$, does not differentiate between the two different types of cells, because one of the classes is assigned to the cells and the other to the background. In other to classify different types of cells we can increase the number of classes to 3.





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

In the implementation of k-means clustering of *Scikit Learn*, the different possibilities for initializing the classes are:

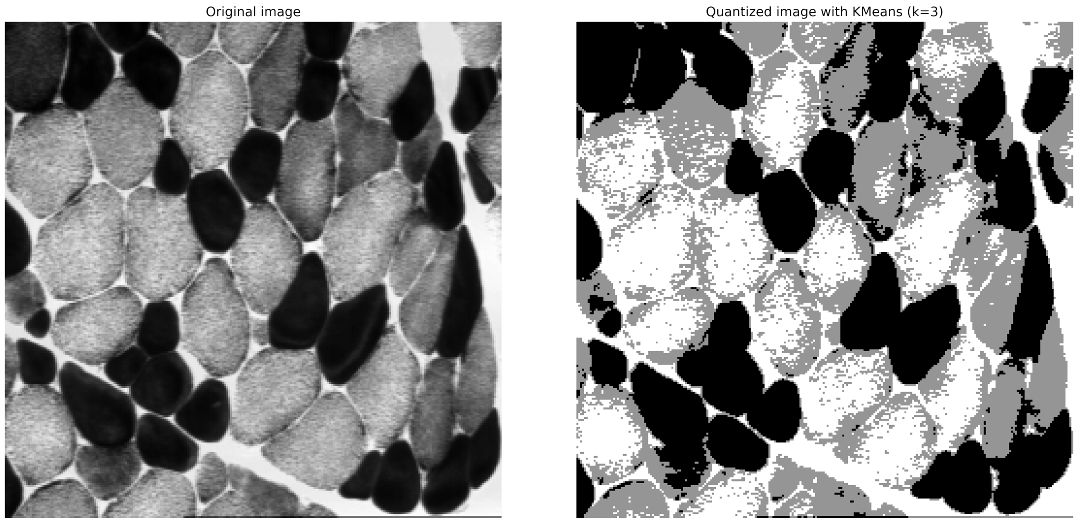
- handpick the initial centers and pass them as arguments to the k-means algorithm;
- choose observations at random from the data as the initial centroids;
- use the "k-means++" technique: an initial center is chosen uniformly at random from the data points, the next center is chosen from the data points with probability proportional to the point's contribution to the overall potential, this step is repeated until k centers have been chosen.

La classification obtnue est-elle stable (même position finale des centres des classes) avec une initialisation aléatoire? Testez sur différentes images à niveaux de gris et différents nombres de classes.

Executing the k-means algorithm multiple times, with random initialization, we observe that the class centers are stable.

Quelles sont les difficultés rencontrées pour la segmentation des différentes fibres musculaires dans l'image `muscle.tif`?

The image `muscle.tif` was segmented into three classes, in a bid to differentiate the white background, the lighter colored fibers and the darker colored ones. Applying the k-means algorithm, we see that it fails to accurately segment the lighter colored fibers, this happens because they have a granular texture, therefore the algorithm fails to identify it as a single object.



Expliquez pourquoi le filtrage de l'image originale (filtre de la moyenne ou filtre median) permet d'améliorer la classification.

Filtering the image prior to classification improves the performance of the k-means algorithm because the objects become more homogeneous; noise and texture are smoothed out.

3.2 Image en couleur

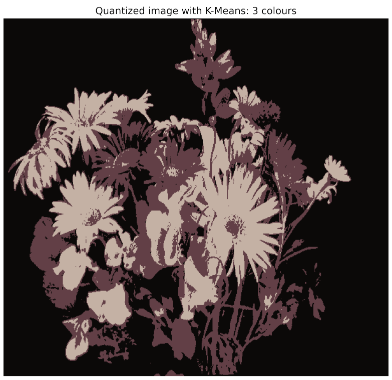
Testez l'algorithme sur l'image `fleur.tif` pour une classification en 10 classes, les centres des classes initiaux étant tirés aléatoirement).

Commentez la dégradation de l'image quantifiée par rapport à l'image initiale.

After quantification, the image is coded with only 10 colors, therefore, even though we can still distinguish the flowers, some information is lost in comparison with the original image, this is evident where we had color gradients, that became discretized. Also, the more frequent colors were kept, while rare ones, such as blue, were lost.

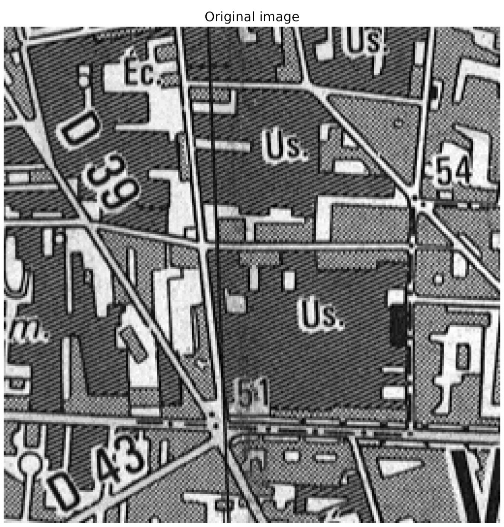
Quel est le nombre minimum de classes qui donne un rendu visuel similaire à celui de l'image codée sur 3 octets?

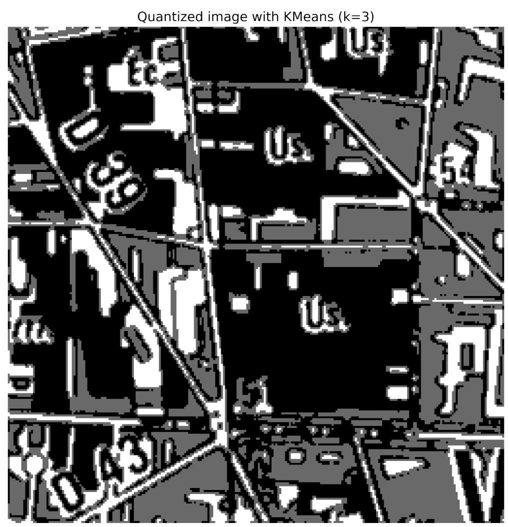
Testing the algorithm for a growing number of classes, we get a result visually similar to the original image for $k = 9$ classes.



Proposez une solution pour retrouver les planches-mères utilisées pour l'impression d'une carte IGN: `carte.tif`.

To find the *planches-mères* of the map `carte.tif`, we could first apply a mean filter to smooth out the crosshatched and dotted textures, making the areas belonging to a same class homogeneous. Then the k-means algorithm could be used to segment the map into three classes.





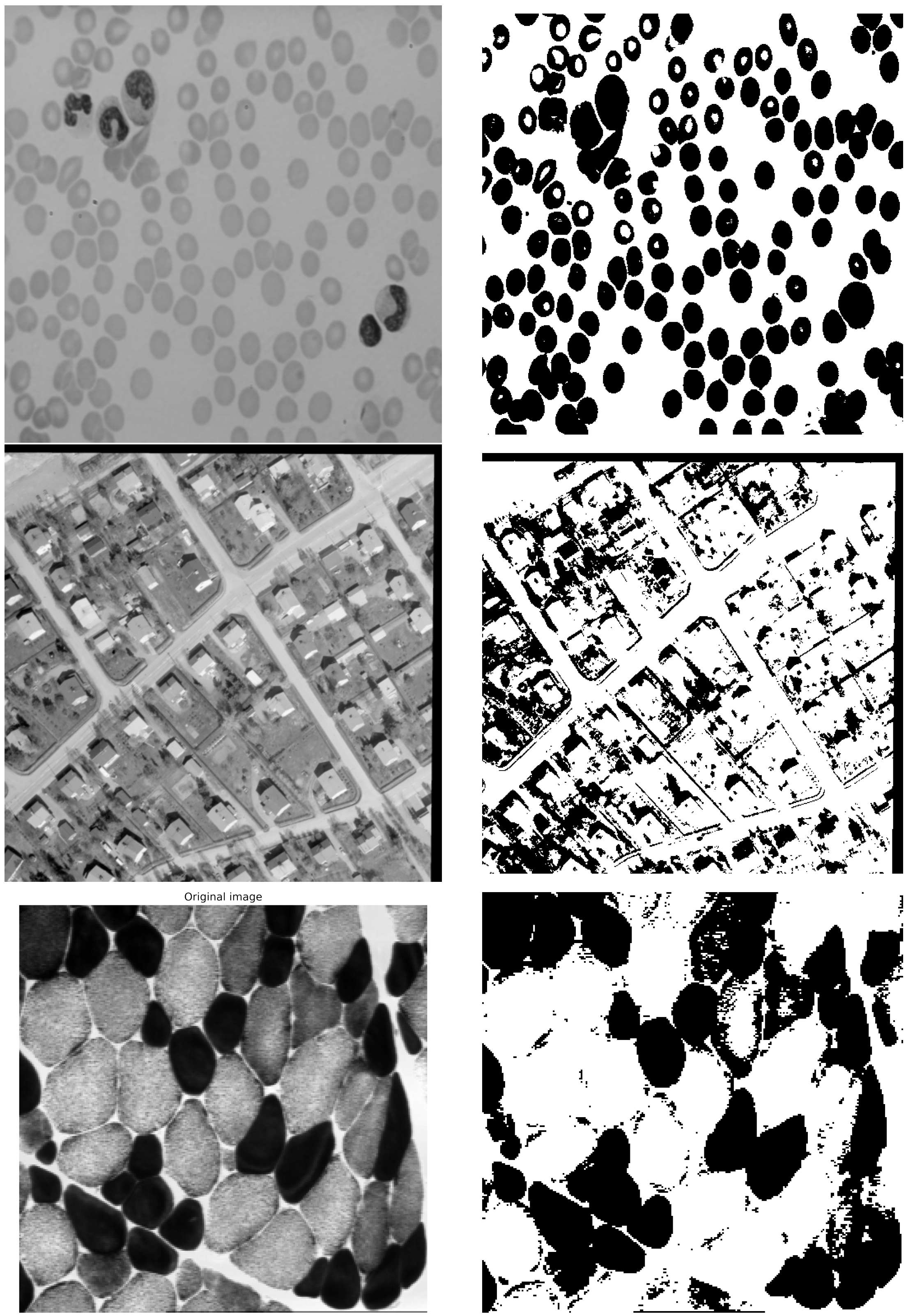
4 Seuillage automatique: Otsu

Dans le script `otsu.py` quel critère cherche-t-on à optimiser?

In the python script the criterion being optimized is the minimization of intra-class variance.

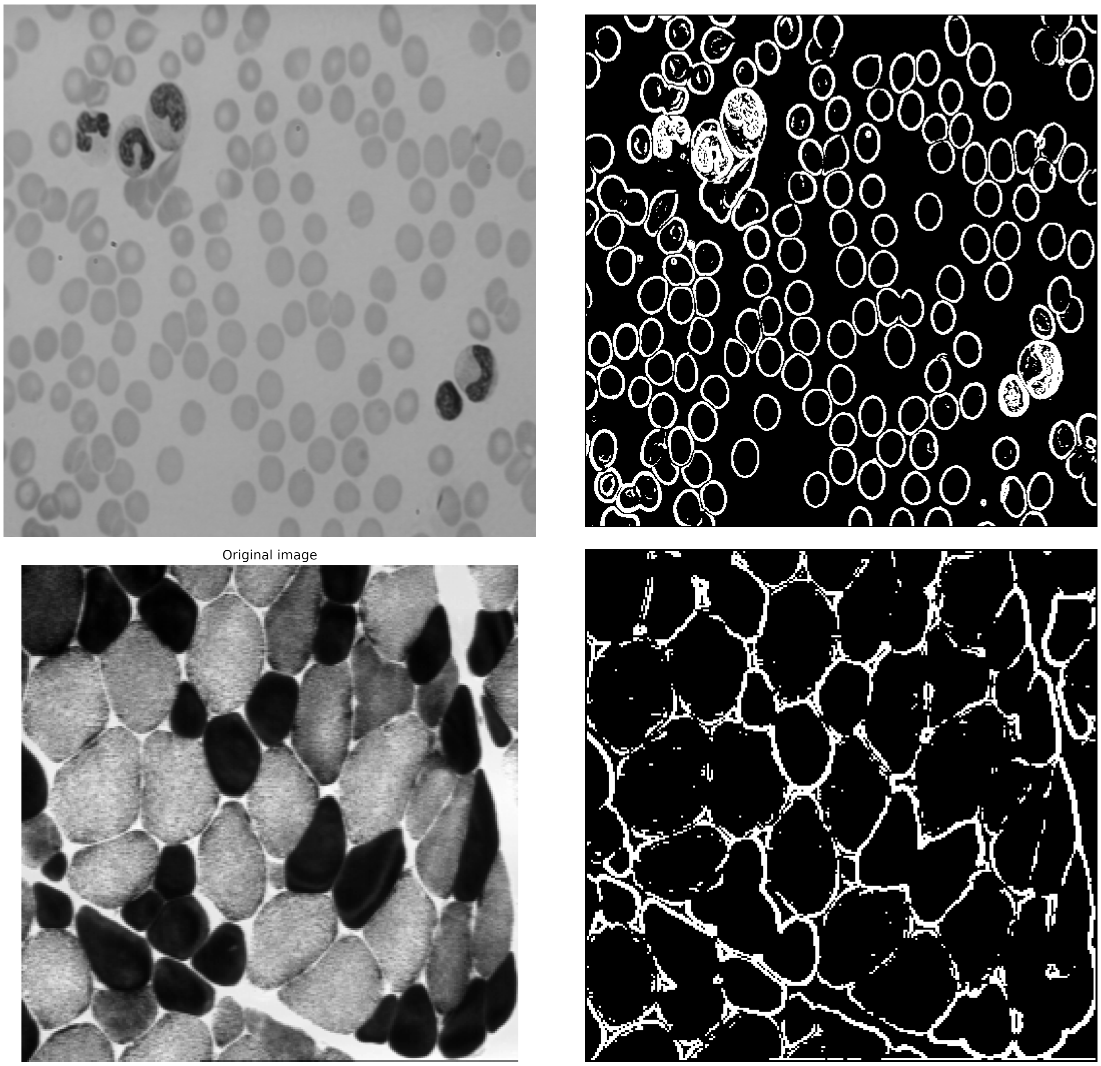
Testez la méthode de Otsu sur différentes images à niveaux de gris, et commentez les résultats.

Applying Otsu's segmentation method for two classes to different gray scale images, we see that it does not always successfully distinguish the objects from the background, this is evident for the image depicting muscle fibers.



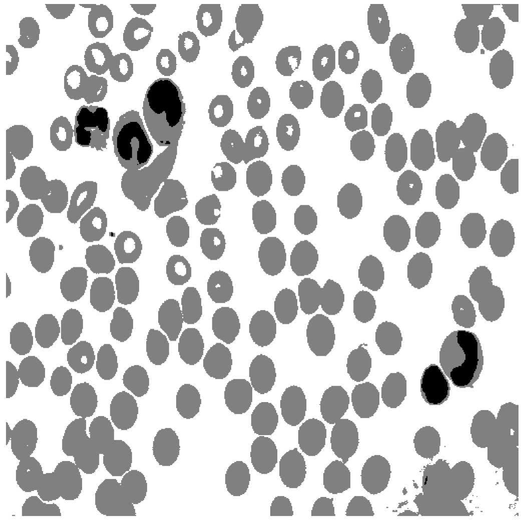
Cette méthode permet-elle de seuiller correctement une image de norme du gradient?

Otsu's method can be successfully used to detect contours by choosing an adequate threshold for the norm of the gradient.



Modifiez ls script `otsu.py` pour traiter le problème à trois classes, i.e. la recherche de deux seuils.

The result of the application of Otsu's method for three classes is illustrated in the image bellow.



```python
def otsu_thresh_modified(im):

    h=histogram(im)

    m=0
    for i in range(256):
        m=m+i*h[i]    # mean

    mint1=0
    mint2=0
    minv=np.inf

    for t1 in range(1,256-2):
        for t2 in range(t1+1, 256-1):

            w0=0; w1=0; w2=0
            m0=0; m1=0; m2=0
            v0=0; v1=0; v2=0

            for i in range(t1):
                w0=w0+h[i]
                m0=m0+i*h[i]
            if w0 > 0:
                m0=m0/w0
            for i in range(t1):
                v0=v0+h[i]*(i-m0)**2
            if w0 > 0:
                v0=v0/w0


            for i in range(t1, t2):
                w1=w1+h[i]
                m1=m1+i*h[i]
            if w1 > 0:
                m1=m1/w1
            for i in range(t1, t2):
                v1=v1+h[i]*(i-m1)**2
            if w1 > 0:
```

```
                v1=v1/w1

            for i in range(t2,256):
                w2=w2+h[i]
                m2=m2+i*h[i]
            if w2 > 0:
                m2=m2/w2
            for i in range(t2,256):
                v2=v2+h[i]*(i-m2)**2
            if w2 > 0:
                v2=v2/w2

            v = w0*v0 + w1*v1 + w2*v2

            if v < minv:
                mint1=t1
                mint2=t2
                minv=v

    return(mint1, mint2)
```

5 Croissance de régions

Quelles contraintes doit vérifier un pixel pour être ajouté à l'objet existant?

In the `region_growing.py` python script, a pixel in the proximity of an existent object is added to it if the local mean calculated on the neighborhood of the pixel is within a given threshold of the local mean of the original germ.

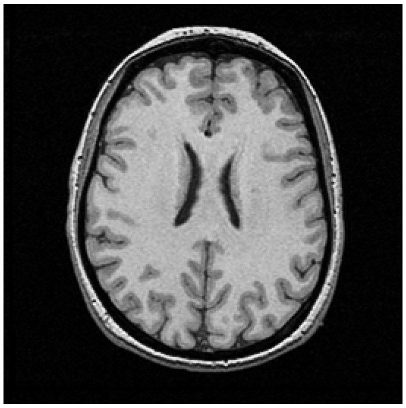
Les paramètres à fixer sont la position du point de départ (x_0, y_0), un seuil *thresh* et le *rayon* qui définit le voisinage sur lequel sont estimés la moyenne et l'écart-type locaux. Quel est l'effet du paramètre *thresh* sur le résultat de segmentation?

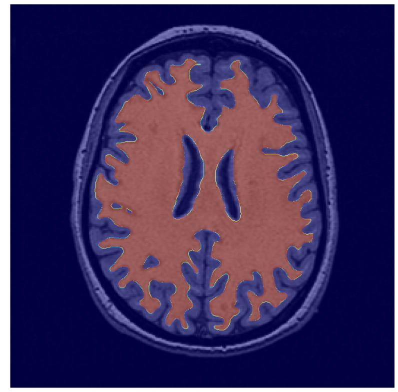
The smaller the threshold the stricter is the predicate, therefore only pixels very similar to the original germ are added to the object, this way, the resulting region might be smaller then the true object and might display unwanted holes, as only slight dishomogeneities will be enough to discard a pixel. Increasing the threshold the predicate becomes milder, allowing a greater growth of the region, if the threshold is too big, the region might exceed the borders of the true object.

Quels paramètres permettent de segmenter correctement la matière blanche?

The following parameters allow the correct segmentation of the white matter:

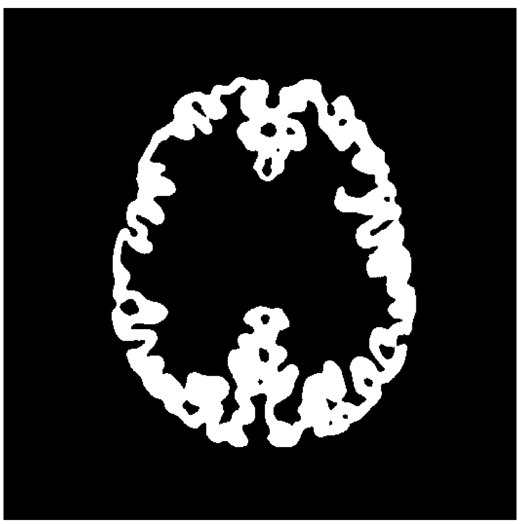
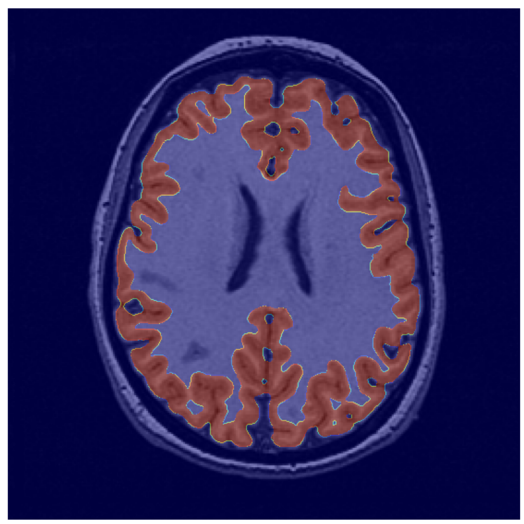
- $(x_0, y_0) = (300, 300)$
- *thresh*$= 4$
- *rayon*$= 5$





Parvenez-vous à segmenter la matière grise également?

Choosing the correct germ, we can segment the gray metter as well, although only the parts connected to the original germ are detected. Disjoint areas of gray matter, such as the ones close to the center of the brain, are not detected.

Quel est le prédicat mis en place dans ce script?

The predicate implemented is “the region is homogeneous”, and the argument used is that the local mean of the neighborhood of each pixel must be within *thresh* times σ of the local mean of the neighborhood of the germ, where σ is the standard deviation of the neighborhood of the germ.

Proposez un autre algorithme qui n’utilise pas la croissance de régions, mais qui donne le même résultat.

Given the initial germ and the mean and standard values of its neighborhood, the algorithm could just verify for each pixel in the image whether its local mean falls within the derired threshold of the local mean of the original germ.

Proposez un prédicat qui nécessite réellement un algorithme de croissance de région.

An adaptation of the predicate that would demand usage of the region growing algorithm would be “the region is homogeneous, connected and contains the initial point”.