

Rapport_Analyse d'Images

——XU Tianjiao && GUO Xiaoqing

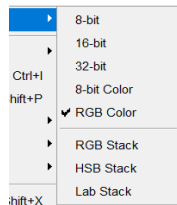
Sommaire

TP1. Prise en main de ImageJ.....	1
Q1. les convertir en images en niveaux de gris.....	1
Q2. Analyse, générez l'histogramme de niveaux de gris.....	2
Q3. Egalisez l'histogramme	2
Q4. Pourquoi, dans la question précédente, l'image soleil est celle semblant avoir le plus "changé" ?	3
Q5. filtres moyenneur	3
Q6. Que donnerait l'application d'un filtre moyenneur dont la dimension serait égale à celle de l'image ?	4
Q7. La succession de convolutions.....	4
Q8. profil d'intensité	5
Q9. les points faibles	6
TP2. Extraction de régions & Détection de contours	7
Q1. le filtre de détection de contours	7
Q2. image-calculator	9
Q3. fore ressortir les traits horizontaux/verticaux	9
Q4. compter le nombre de cellules.....	10
Q5. le moins de cellules possibles restent collées	11
TP3. Création de Plugins sous ImageJ	13
Q2. Expliquez l'algorithme notamment :.....	13
Q3. Compilez et exécutez.....	13
Q4. Calculer la moyenne des niveaux de gris d'une image	13
Q5. Quelle image parmi celles d'un répertoire ressemble le plus à l'image ouverte avec imageJ	14
Q6. Un plugins associant des mot-clés décrivant le contenu.....	14

TP1. Prise en main de ImageJ

Q1. les convertir en images en niveaux de gris

1. les convertir en images en niveaux de gris



8-bit Converts to 8-bit grayscale. ✓

16-bit Converts to unsigned 16-bit grayscale. ✓

32-bit Converts to signed 32-bit floating-point grayscale. ✓

RGB Color Converts to 32-bit RGB color.

RGB Stack Converts to a 3--slice (red, green, blue) stack. The active image must be RGB color.

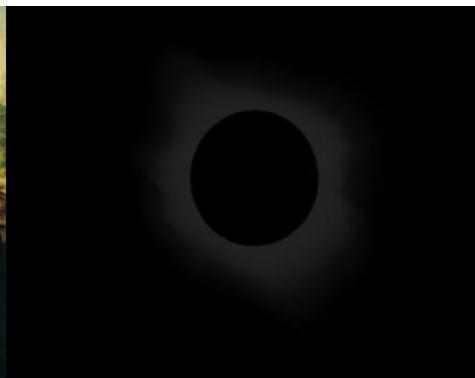
HSB Stack Converts to a 3--slice (hue, saturation and brightness) stack. The active image must be RGB color.

Exemples:

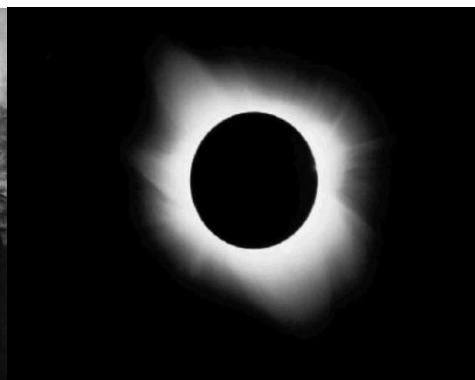
8-bit (lisa)



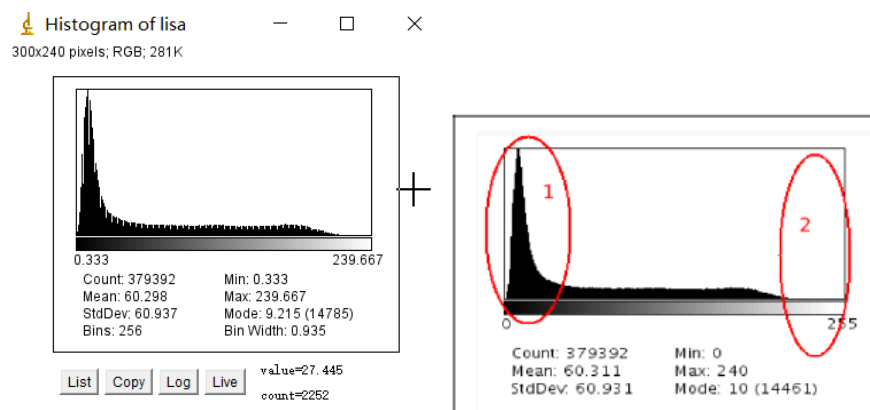
16-bit (soleil)



32-bit (vache)



Q2. Analyse, générez l'histogramme de niveaux de gris



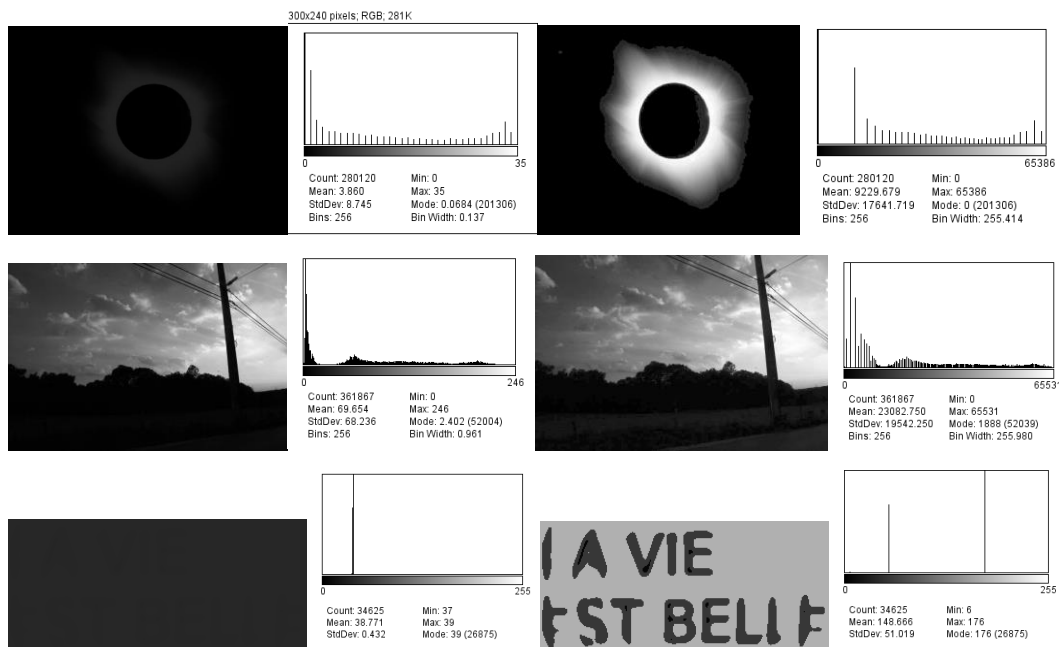
Commentaire: Il décrit le nombre de pixels dans l'image qui ont ce niveau de gris. L'abscisse est le niveau de gris et l'ordonnée est la fréquence à laquelle le niveau de gris apparaît.

point 1: beaucoup de parties sombres

point 2: peu de parties lumineuses

Q3. Egalisez l'histogramme

Image avant / Histogramme de niveaux de gris / Égaliser / Histogramme de niveaux de gris après égalisation



Égaliser l'Histogram: utilise un algorithme modifié qui prend la racine carrée des valeurs de l'histogramme.

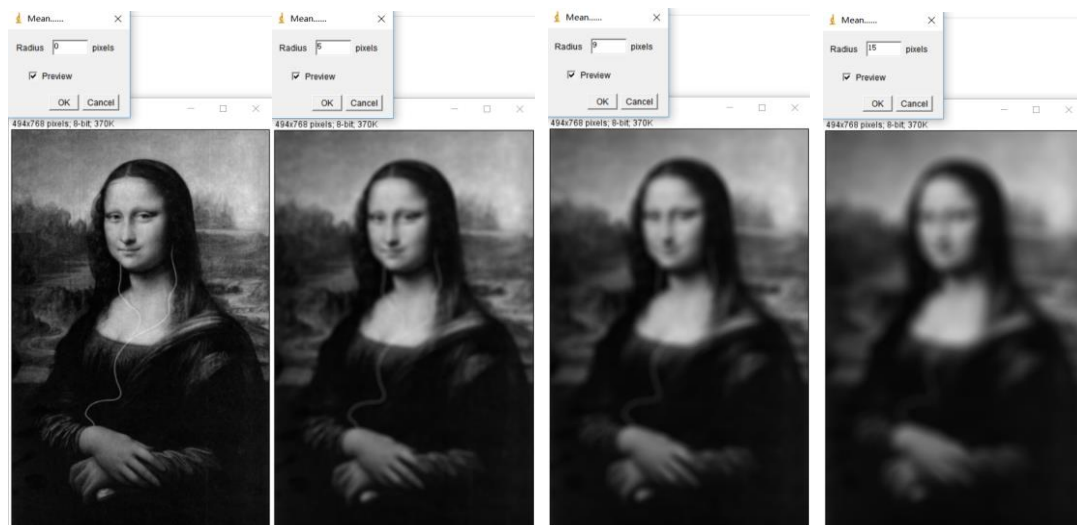
Q4. Pourquoi, dans la question précédente, l'image soleil est celle semblant avoir le plus "changé" ?

Le contraste de l'image est amélioré en utilisant les gris (l'histogramme de niveaux de gris).

L'égalisation de l'histogramme rend la répartition de l'image en niveaux de gris plus équilibrée, c'est-à-dire qu'il va élargir les niveaux de gris qui ont plus de pixels et va compresser les niveaux de gris qui ont moins de pixels, pour étendre la plage dynamique, améliorer le contraste et le changement des niveaux de gris, et rendre l'image plus claire.

La valeur maximale en niveaux de gris de l'image "soleil" d'origine est 35 (l'abscisse de l'histogramme), après l'égalisation, la valeur est 65386 (l'abscisse de l'histogramme égalisé). Pour le comparer avec deux autres images, "paysage" (246 -> 65531) et "mystere" (255 -> 255), l'image "soleil" élargit la plage de niveaux de gris très fortement. Il élargit un grand nombre de niveaux de gris précédemment concentrés, et équilibre et décompose chaque niveau de gris de l'image, donc le contraste et le changement des niveaux de gris est grand après l'égalisation, et l'image est plus claire. Donc l'image "soleil" est celle semblant avoir le plus "changé".

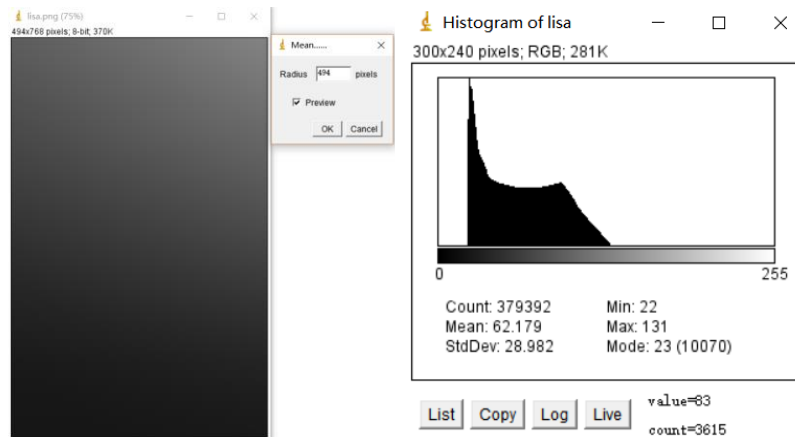
Q5. filtres moyenneur



Le filtre moyenneur permet de lisser l'image (smoothing) -- Remplace chaque pixel par la valeur moyenne de ses voisins. Filtre dont tous les coefficients sont égaux.

Plus la valeur de la taille du filtre est grande, plus l'image est floue.

Q6. Que donnerait l'application d'un filtre moyenneur dont la dimension serait égale à celle de l'image ?



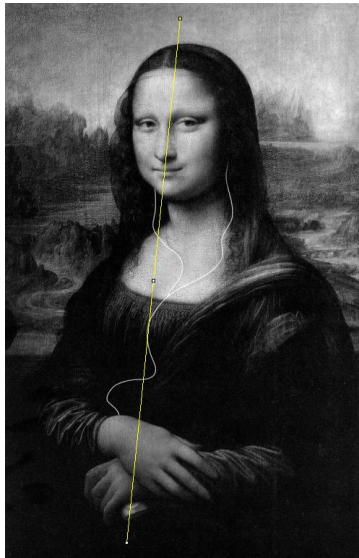
Le fonctionnement est principale de reduire les détails presque indépendants dans l'image, c'est-à-dire la zone de pixel qui est plus petite que la taille de la taille(masque) de filtre. Donc, lorsque la taille du filtre est égale / supérieure à la dimension de l'image, l'image est complètement floue (les niveaux de gris sont environ complètement concentrés).

Q7. La succession de convolutions

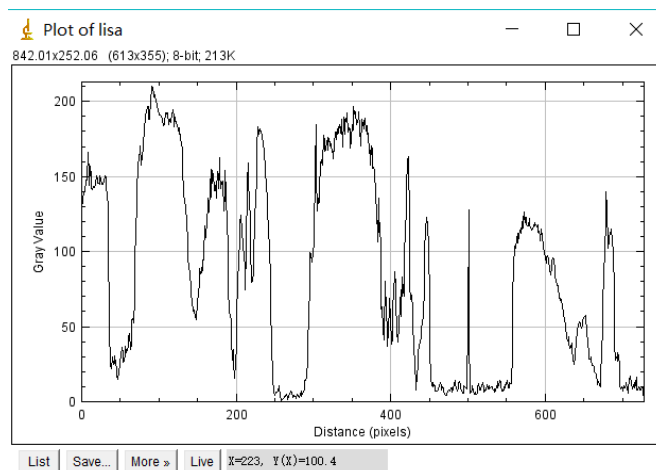


Résultats: les degrés flous d'image sont différent, et l'effet de 2 convolution successives est inférieur à celui de seul filtre moyenneur.

Q8. profil d'intensité



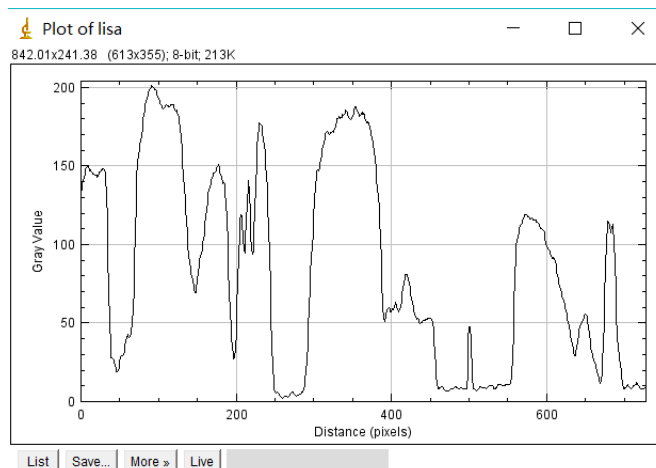
Appliquer Analyse/Plot Profile :



Displays a two-dimensional graph of the intensities of pixels along a line or rectangular selection.

The X-axis represents distance along the line and the Y-axis is the pixel intensity.

nouveau profil:



La courbe est plus douce et lisse.

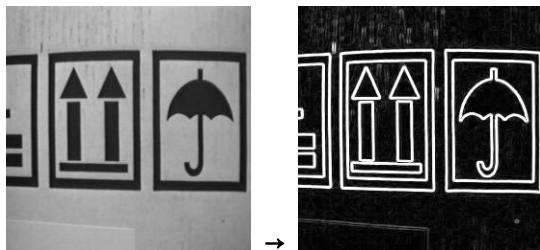
Q9. les points faibles

Si le noyau de convolution n'est pas centré-symétrique, les opérations de convolution et de filtrage obtiendrez des résultats complètement différents. Les opérations de convolution peuvent faire en sorte que l'image devienne plus petite (arêtes d'image de perte).

TP2. Extraction de régions & Détection de contours

Q1. le filtre de détection de contours

Sobel :

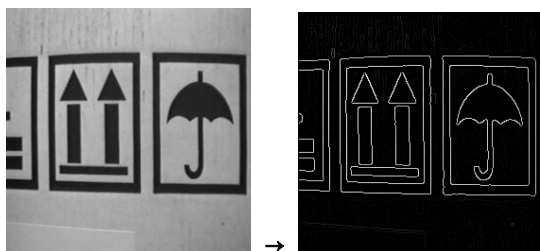


Utilise un détecteur de bord Sobel pour mettre en évidence des changements brusques d'intensité dans l'image ou la sélection active.

Algorithme: Deux noyaux de convolution 3×3 (illustrés ci-dessous) sont utilisés pour générer des dérivées verticales et horizontales. L'image finale est produite en combinant les deux dérivés en utilisant la racine carrée de la somme des carrés.

1	2	1	1	0 - 1
0	0	0	2	0 - 2
- 1	- 2	- 1	1	0 - 1

Deriche:



"Deriche2_.java" effectue un filtrage de Canny-Deriche pour la détection de bord.

Un paramètre contrôle le degré de lissage appliqué; la valeur par défaut est 1.0, des valeurs plus élevées impliquent moins de lissage mais une détection plus précise, des valeurs plus basses impliquent plus de lissage mais une détection moins précise. Une suppression non maximale est ensuite effectuée pour obtenir des bords fins.

L'angle des arêtes peut être affiché (par défaut: pas choisir).

Traitement - aqui:

Image initiale / Sobel (Find Edges) / Deriche



Sobel:

On exécute "Find Edges" par 3 fois et donc il montre un fort contraste entre le bord de l'image et le reste de l'image.

Deriche:

Nous fixe la valeur à 10 et il implique moins de lissage mais une détection plus précise. On peut obtenir un nouveau image avec des bords fins.

Le méthode Deriche est meilleur pour la détection de contours sur cette image.

Traitement - noise:

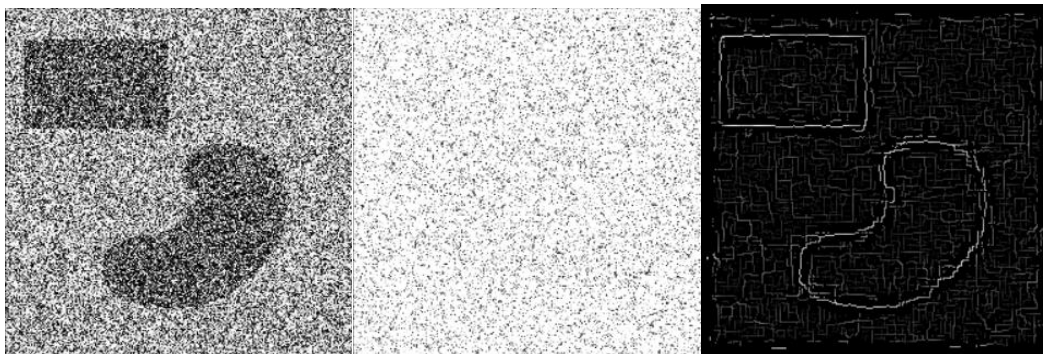
Image initiale

/

Sobel (Find Edges)

/

Deriche



Sobel:

On exécute "Find Edges" par 2 fois mais et il y a plus de bruit et les bords de l'image d'origine ont disparu.

Deriche:

Nous fixe la valeur à 0.2 et il implique plus de lissage mais une détection moins précise. Mais il y a beaucoup de bruit dans cette image, certainement ça marche bien. On peut obtenir un nouveau image avec des bords fins.

Le méthode Deriche est meilleur pour la détection de contours sur cette image.

Traitement - delphin:

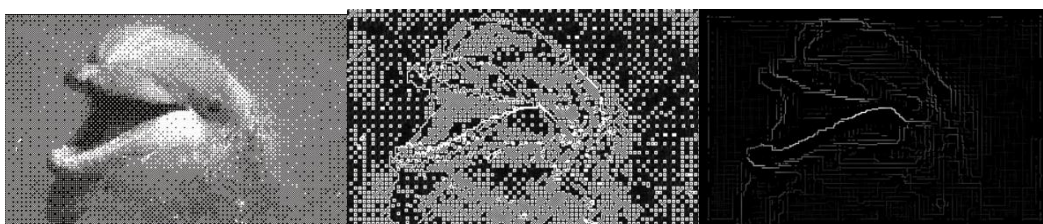
Image initiale

/

Sobel (Find Edges)

/

Deriche



Sobel:

On exécute "Find Edges" par 2 fois mais il y a plus de bruit mais les bords de l'image étaient flous.

Deriche:

Nous fixe la valeur à 0.2 et il implique plus de lissage mais une détection moins précise. Mais il y a beaucoup de bruit dans cette image, mais les bords de l'image sont encore un peu flous.

Le méthode Deriche est relativement meilleur pour la détection de contours sur cette image.

Q2. image-calculator

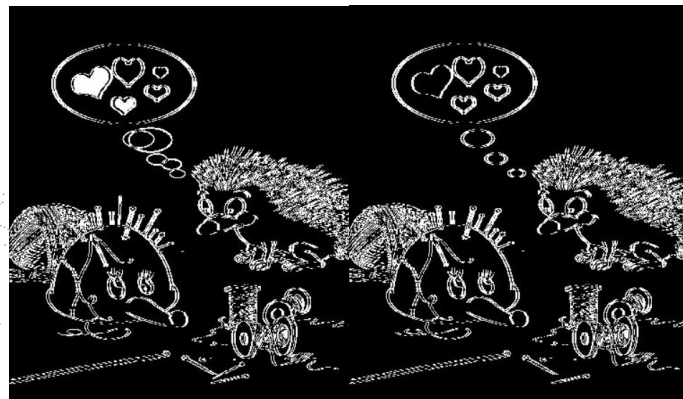
L'algorithme : XOR ($img1 = img1 \oplus img2$) ou Difference ($img1 = |img1 - img2|$)

Les images à comparer et les resultats:

jeu1 et jeu2



jeu1 et jeu3



jeu2 et jeu3

Q3. fore ressortir les traits horizontaux/verticaux

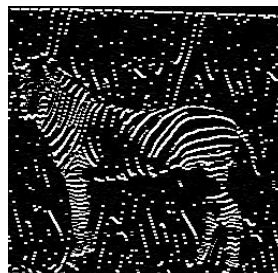
Filtre de Prewitt

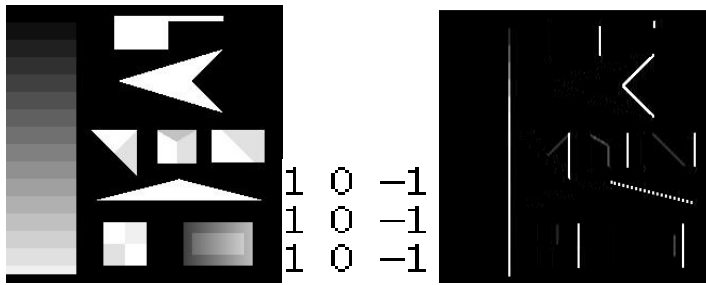
Étapes: Process -> Filters -> Convolve ...

fore ressortir les traits horizontaux:



fore ressortir les traits verticaux:

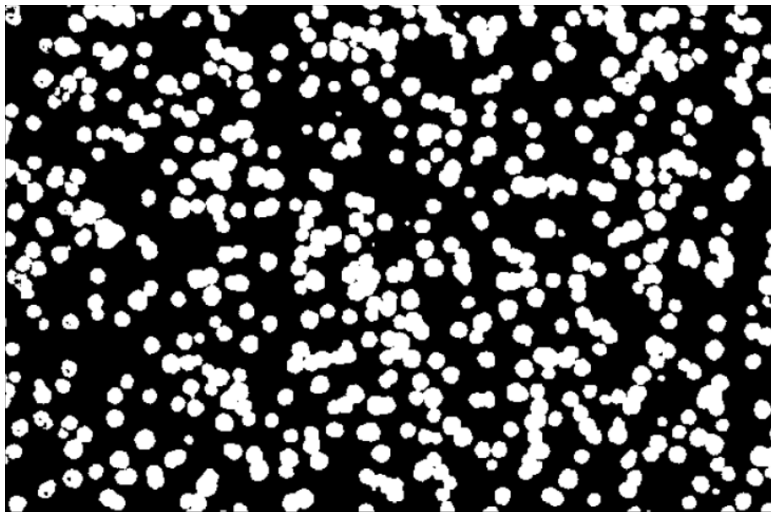




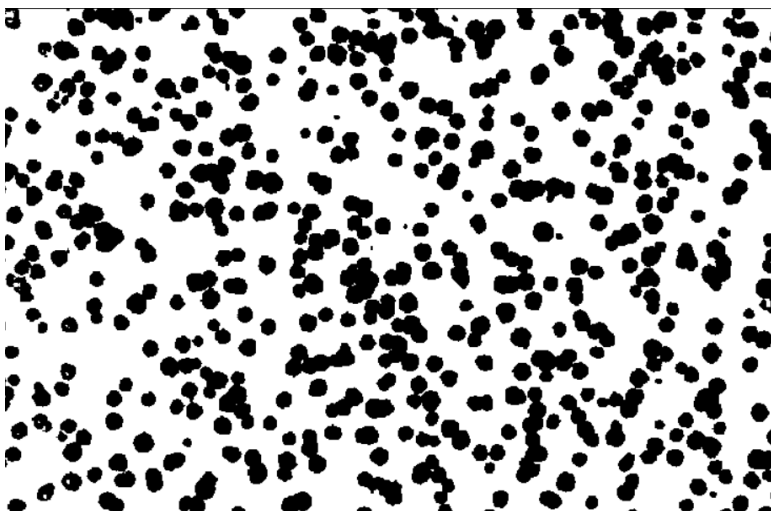
Q4. compter le nombre de cellules

Chargez l'image cellule.png et binarisez là.

Image cellule:

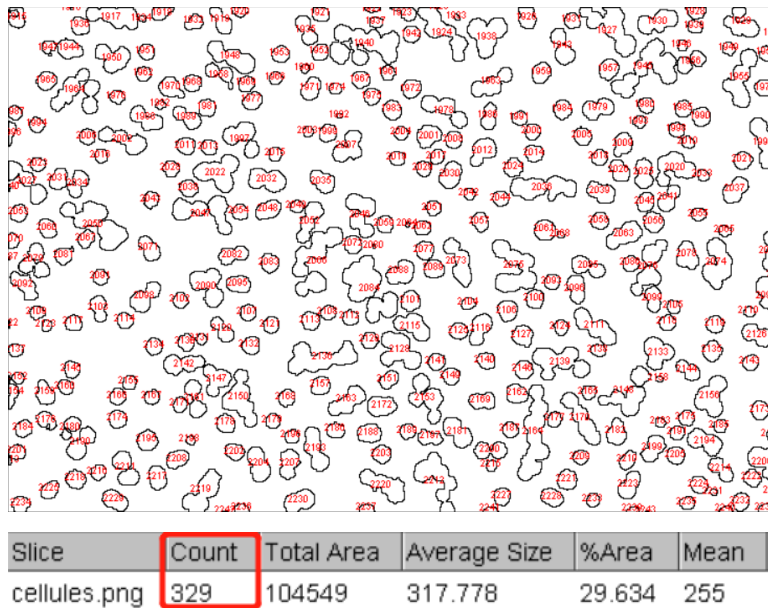


Binarize image: Process -> Binary -> Make Binary



Vérifiez que imageJ indique bien la présence de 329 cellules.

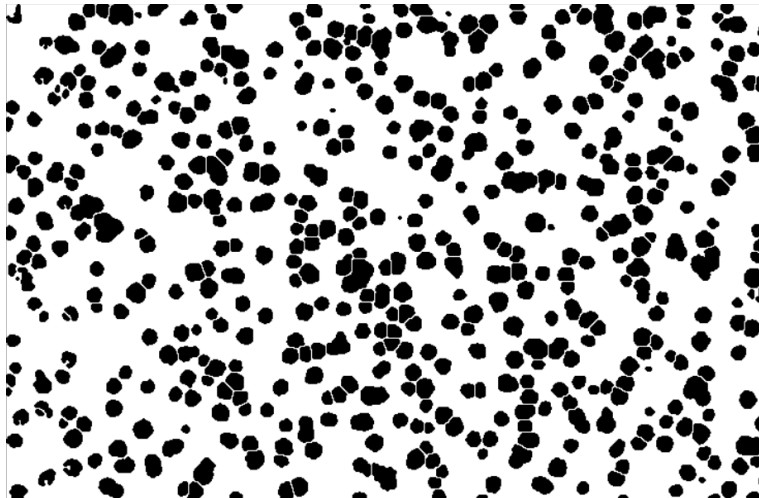
Counter cellules: Analyze -> Analyze Particles -> "Size: 0 + Show: Outlines + Summarize"



Q5. le moins de cellules possibles restent collées

Traitement: Des cellules sont collées sur la photo alors qu'elles sont dissociées dans la réalité

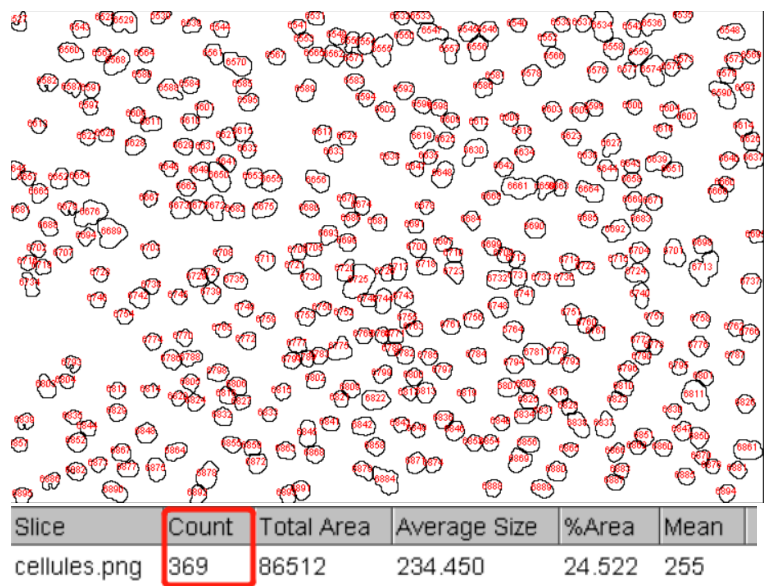
Process -> Binary -> Watershed



Traitement: Des petites taches noires sont apparues un peu partout et sont interprétées comme étant des cellules

Options -> "Count: 5 + Do: Erode"

Counter cellules: Analyze -> Analyze Particles -> "Size: 147 + Show: Outlines + Summarize"



TP3. Création de Plugins sous ImageJ

Q2. Expliquez l'algorithme notamment :

Quelle est la taille du tableau `pixels[]` ? Donnez une formule générale permettant d'accéder à n'importe quel pixel d'une image via le tableau `pixels` : `ndg[x][y]=pixels[???]`

La taille de la table de pixels est le nombre de pixels de l'image (`height * width`).

`ndg[x][y]=pixels[y * width + x]`

Q3. Compilez et exécutez



Il binarise l'image.

Q4. Calculer la moyenne des niveaux de gris d'une image

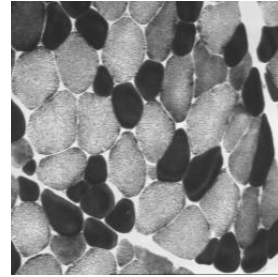
```
int sum = 0 ;  
for(int i = 0 ; i < pixels.length ; i++){  
    sum += pixels[i];  
}
```

```
int avg = sum / pixels.length;
```

Q5. Quelle image parmi celles d'un répertoire ressemble le plus à l'image ouverte avec imageJ



Le plus ressemble aux niveaux de gris:



Message



L'image la plus proche est D:\Polytech\Images\Images\musc.jpg avec une distance de 0.89959716796875

OK

Q6. Un plugin associant des mots-clés décrivant le contenu

couleur, contenu, qualité