


# Compte-rendu du TP4 - Traitement d'images

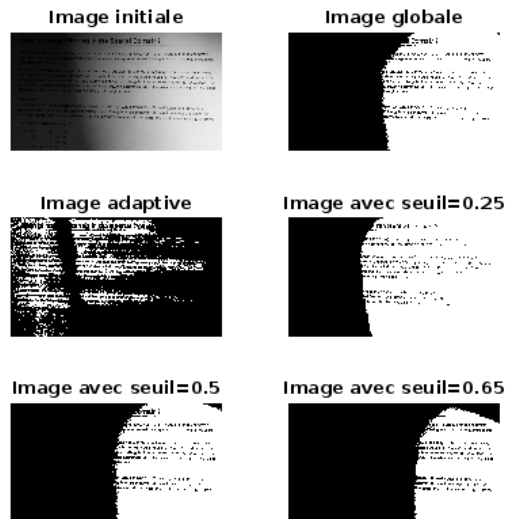
✓ Réalisée par: Roudayna Lazreg - Groupe G / sous-groupe: 2

## Exercice 1:

-  1. Ecrivez un programme Matlab permettant d'afficher sur la même figure :
  - L'image 'printedtext.png' ayant pour titre 'Image\_Initiale'.
  - Cinq images résultantes de la binarisation de l'image initiale globalement, de façon adaptative et en utilisant trois seuils fixes manuellement (les seuils doivent être entre 0 et 1). Chaque image doit avoir un titre par la transformation subie.
2. Commenter les résultats obtenus.
3. En utilisant la même image appliquer une binarisation adaptative en utilisant le paramètre "ForegroundPolarity" pour indiquer que le premier plan est plus sombre que l'arrière-plan et en spécifiant la "sensitivity" à 0.4. Que remarquez-vous ?

## Réponse:

```
1 clear all
2 close all
3 %%Q1:
4 i=imread('printedtext.png');
5 subplot(321)
6 imshow(i);title("Image initiale");
7
8 bwg=imbinarize(i,"global");
9 subplot(322)
10 imshow(bwg);title("Image globale")
11
12 bwa=imbinarize(i,"adaptive");
13 subplot(323)
14 imshow(bwa);title("Image adaptive")
15
16 bw025=imbinarize(i,0.25);
17 subplot(324);
18 imshow(bw025);title("Image avec seuil=0.25")
19
20 bw05= imbinarize(i,0.5);
21 subplot(325);
22 imshow(bw05);title("Image avec seuil=0.5")
23
24 bw065= imbinarize(i,0.65);
25 subplot(326);
26 imshow(bw065);title("Image avec seuil=0.65");
27 %%Q2: on constate que l'image adaptative ici est la plus efficace pour le Seuilage d'une image.
```



```

1 %%Q3:
2 figure;
3 ba=imbinarize(i,"adaptive","ForegroundPolarity","dark","Sensitivity",0.4);
4 imshow(ba);

```

### What Is Image Filtering in the Spatial Domain?

Filtering is a technique for modifying or enhancing an image. For example, you can blur an image to emphasize its general features or remove other features. Image smoothing operations are performed with filters such as blurring, sharpening, and edge enhancement.

Filtering is a neighborhood operation. Instead of using pixel values in the entire image to determine an output pixel, some algorithm uses the values of the pixels in the neighborhood of the corresponding input pixel. A filter is defined as a set of pixels, defined by their locations relative to that pixel. (See the neighborhood of Pixel P in Figure 10.10.1 for a general discussion of neighborhood operations.) Linear filtering is filtering in which the value of an output pixel is a linear combination of the values of the pixels in the neighborhood neighborhood.

#### Convolution

Linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of the gray-level input pixels. The matrix of weights is called the convolution kernel. It also is known as the filter. A convolution kernel is a symmetric kernel built as both a column and a row.

For example, suppose the image is

```

A = [17 24 1 8 15
     23 5 7 34 36
     4 5 12 28 22
     10 22 18 21 3
     4 4 2 1 1]

```

## Exercice 2: [↗](#)

1. Ecrivez un programme Matlab permettant d'afficher sur la même figure :
  - L'image 'foggysf2.jpg' ayant pour titre 'Image\_Initiale'.
  - L'image en niveaux de gris.
  - L'histogramme de l'image.
  - Le résultat de binarisation de l'image par la méthode globale.
2. Selon l'histogramme de l'image identifiez approximativement le nombre de seuils à choisir pour segmenter l'image.
3. Utilisez ce nombre pour segmenter l'image et affichez le résultat dans une nouvelle fenêtre (Utiliser la fonction label2rgb pour convertir l'image segmentée en une image couleur).
4. Augmenter le nombre de seuils afin d'améliorer le résultat de segmentation.

## Réponse: [↗](#)

```

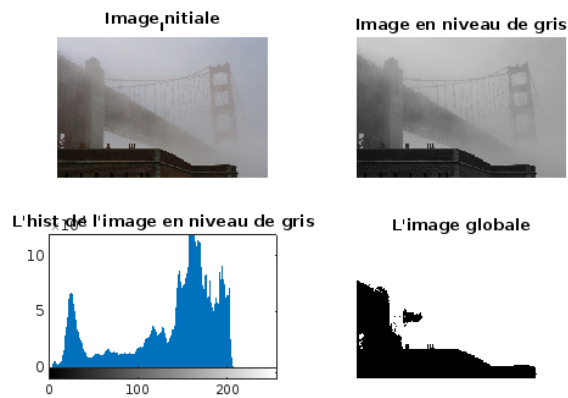
1 clear all
2 close all

```

```

3 %%Q1:
4 clear all
5 close all
6 figure;
7 i2=imread('foggysf2.jpg');
8 subplot(221)
9 imshow(i2) ;title('Image_Initiale');
10
11 %%L'image en niveaux de gris
12 g=rgb2gray(i2);
13 subplot(222)
14 imshow(g);title('Image en niveau de gris');
15
16 %%L'histogramme de l'image.
17 subplot(223)
18 imhist(g);title("L'hist de l'image en niveau de gris");
19
20 %%Le résultat de binarisation de l'image par la méthode globale.
21 b=imbinarize(g,"global");
22 subplot(224)
23 imshow(b);title("L'image globale");

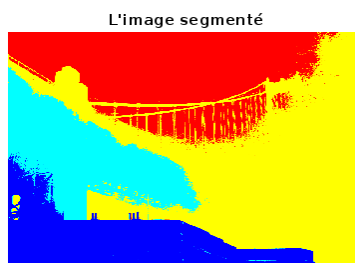
```



```

1 %%Q2:
2 nombre de seuils =3
3 %%Q3;
4 s=multithresh(g,3);
5 q=imquantize(g,s);
6 seg=label2rgb(q);
7 figure;
8 imshow(seg)

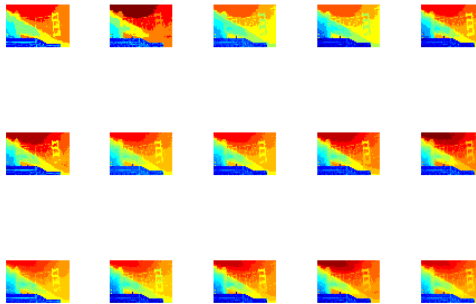
```



```

1 figure;
2 %%Q4:
3 for n=6:20
4 s=multithresh(g,n);
5 q=imquantize(g,s);
6 seg=label2rgb(q);
7 subplot(3,5,n-5),
8 imshow(seg),
9 end

```



### Exercice 3: [🔗](#)

1. Ecrivez un programme Matlab permettant de lire l'image 'medtest.png' (voir le fichier Zip [region\\_growing](#)), de l'affecter à la variable I et de l'afficher.
2. Convertissez l'image en double précision en utilisant la fonction `im2double` et noter la variable `I_double`.
3. Choisir comme point germe le pixel de coordonnées `x=120` et `y=260` ;
4. Segmentez l'image par application de l'algorithme de croissance de région autour du point germe de coordonnées (x,y) et varier le critère d'homogénéité (écart du niveau d'intensité maximal) avec les valeurs 0.05, 0.1 et 0.2 autour du point germe [x y]. Noter les trois images J1, J2, et J3.
5. Créer une fonction qui calcule la surface de la région segmentée : `S=area(I_double)`
6. Affichez sur une même figure, les images I, J1, J2 et J3 avec la surface calculée dans le titre comme le montre la figure suivante.

### Réponse: [🔗](#)

```

1 clear all
2 close all
3 figure;
4 %%Q1:
5 i3=imread('medtest.png');
6 subplot(221);imshow(i3);title("image initiale")
7
8 %%Q2:
9 c=im2double(i3);
10
11 %%Q3:
12 x=120;y=260 ;
13

```

```

14 %%Q4:
15 j1=regiongrowing(c,x,y,0.05);
16 s1=surfacee(j1);
17 subplot(222);
18 imshow(j1);title("max dist=0.05, S= "+s1);
19 j2=regiongrowing(c,x,y,0.1);
20 s2=surfacee(j2);
21 subplot(223);
22 imshow(j2);title("max dist=0.1, S= "+s2)
23 j3=regiongrowing(c,x,y,0.2);
24 s3=surfacee(j3);
25 subplot(224);
26 imshow(j3);title("max dist=0.2, S= "+s3)

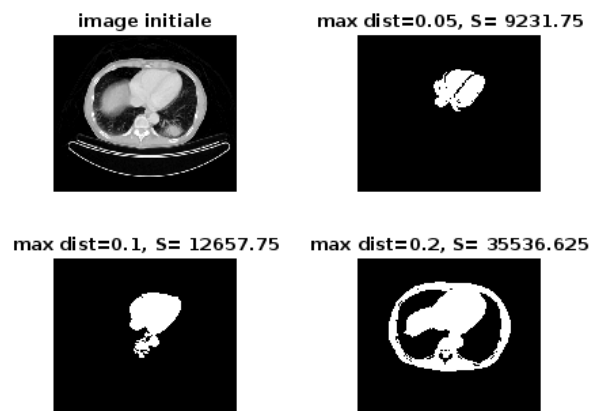
```

⚠ Cette fonction doit être implémentée à la fin de script ou dans un autre script apart.

```

1 %%Q5:
2 function s=surfacee(i)
3 s=bwarea(i);
4 end
5 %%Q6:affichage

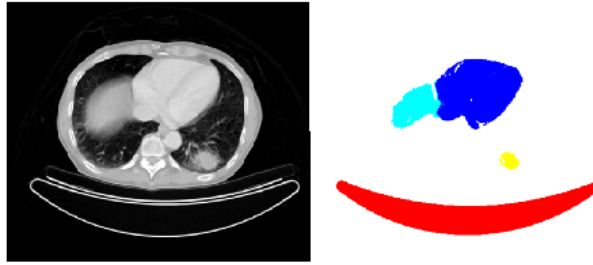
```



```

1 %%Q7:
2 js=zeros(size(c));
3 G=[120 260;160 160;248 310;340 240];
4 for i=1:size(G,1)
5 x=G(i,1);y=G(i,2);
6 j=regiongrowing(c,x,y,0.07);
7 js=js+j*i;
8 s(i)=surfacee(j);
9 end
10 res=label2rgb(js);
11 figure;
12 montage({c,res});

```



```
1 disp("surface de region");
2 disp(s');
```

surface de region

1.0e+04 \*

1.0313

0.3233

0.0566

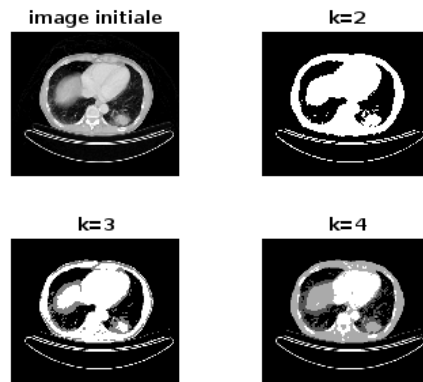
1.4774

#### Exercice4: [↗](#)

1. Ecrivez un programme Matlab permettant d'afficher sur la même figure :
  - L'image 'medtest.png' ayant comme titre 'Image initiale.
  - Le résultat de segmentation de cette image en choisissant comme nombre de cluster 2, 3 et 4.
2. Comparez le résultat obtenu par celui de la méthode de croissance des régions.
3. Refaites le même travail avec l'image 'foggysf2.jpg' et comparez le résultat obtenu avec celui de l'exercice 2 (il faut transformer l'image en niveaux de gris avant la segmentation).
4. Essayez de segmenter une image RGB de votre choix et précisez le nombre de cluster final à adopter.

#### Réponse: [↗](#)

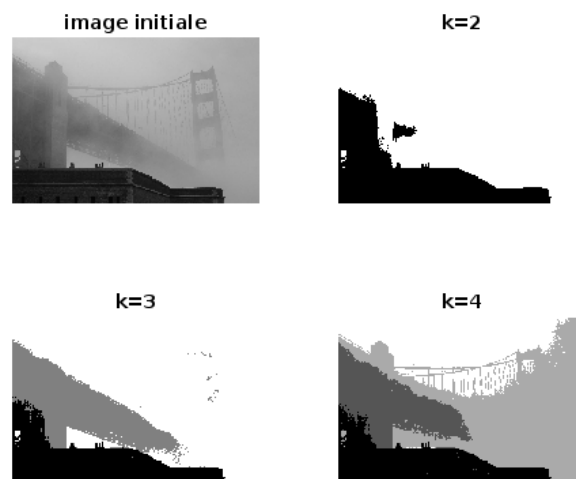
```
1 clear all
2 figure;
3 %%Q1:
4 i4=imread('medtest.png');
5 subplot(221);imshow(i4);title("image initiale")
6
7 [mu1,mask1]=kmeans(i4,2);
8 %%convert matrix to image
9 mask1=mat2gray(mask1);
10 subplot(222);imshow(mask1);title("k=2")
11
12 [mu2,mask2]=kmeans(i4,3);
13 mask2=mat2gray(mask2);
14 subplot(223);imshow(mask2);title("k=3") ;
15
16 [mu3,mask3]=kmeans(i4,4);
17 mask3=mat2gray(mask3);
18 subplot(224);imshow(mask3);title("k=4")
19
```



```

1 clear all
2 figure;
3 %%Q3:
4 i=imread('foggysf2.jpg');
5 i4=rgb2gray(i);
6 subplot(221);imshow(i4);title("image initiale")
7
8 [mu1,mask1]=kmeans(i4,2);
9 mask1=mat2gray(mask1);
10 subplot(222);imshow(mask1);title("k=2")
11
12 [mu2,mask2]=kmeans(i4,3);
13 mask2=mat2gray(mask2);
14 subplot(223);imshow(mask2);title("k=3") ;
15
16 [mu3,mask3]=kmeans(i4,4);
17 mask3=mat2gray(mask3);
18 subplot(224);imshow(mask3);title("k=4")

```



```

1 %%4:
2 clear all
3 figure;
4 i4=imread('football.jpg');

```

```

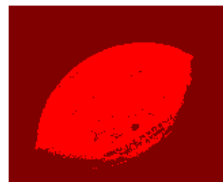
5 subplot(221);imshow(i4);title("image initiale")
6
7 [mu1,mask1]=kmeans(i4,2);
8 mask1=mat2gray(mask1);
9 subplot(222);imshow(mask1);title("k=2")
10
11 [mu2,mask2]=kmeans(i4,3);
12 mask2=mat2gray(mask2);
13 subplot(223);imshow(mask2);title("k=3") ;
14
15 [mu3,mask3]=kmeans(i4,4);
16 mask3=mat2gray(mask3);
17 subplot(224);imshow(mask3);title("k=4")
18 %%k=3 je pense est le meilleur choix

```

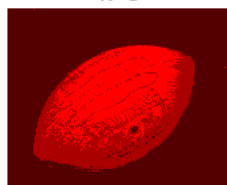
image initiale



k=2



k=3



k=4

