CRTP: Imagerie

BAKHOUCH AYOUBE

Introduction au traitement d'images
CRTP

MA0942

Bakhouch Ayoube

Février 2018

Contents

Chapter 1

Introduction

le compte rendu contient les scripts des algorithmes de chaque fonctions demandée ainsi que les résultats, pour les fonctions et leurs inverses, ils ont été testé sur une image et l'image obtenu (teste de la fonction inverse), pour bien verifier que ca donnait des resultats cohérents.
Les scripts des resultats permets de connaitre les differents placements des resultats images obtenus.
le fichier main contient tous les testes effectué.
J'ai departagé les TP en 2 Sections.

```
clear all
image=imread('100.jpg');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1);
%imshow(image);
%subplot(2,2,2);
%imshow(rgb2gray1(image));
%subplot(2,2,3);
%imshow(rgb2gray2(image));
%subplot(2,2,4);
%imshow(rgb2gray3(image));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1);
%imshow(image);
%subplot(2,2,2);
%imshow(rgbtonrgb1(image));
%subplot(2,2,3);
%imshow(rgbtonrgb2(image));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%image2=III(image);
%image3=III_inv(image2);
%subplot(2,2,1),imshow(image)
%subplot(2,2,2),imshow(image2);
%subplot(2,2,3),imshow(image3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%image2=rgbtoyiq(image);
%image3=yiqtorgb(image2);
%subplot(2,2,1),imshow(image);
%subplot(2,2,2),imshow(image2);
%subplot(2,2,3),imshow(image3);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%image2=rgbtoyuv(image);
%image3=yuvtorgb(image2);
%subplot(2,2,1),imshow(image);
%subplot(2,2,2),imshow(image2);
%subplot(2,2,3),imshow(image3);
```

Chapter 2

Partie I

2.1 RGB-to-GRAY

2.1.1 RGB-to-GRAY-1

```
function grayImage = rgb2gray1(rgbImage)
try
  [rows, columns, numberOfColorChannels ]= size(rgbImage);

 redChannel =uint16( rgbImage(:, :, 1));
 greenChannel =  uint16(rgbImage(:, :, 2));
 blueChannel = uint16( rgbImage(:, :, 3));
 grayImage = (double(redChannel) + double(greenChannel) +double(blueChannel))/3;
 grayImage = uint8(grayImage);


end
```

2.1.2 RGB-to-GRAY-2

```
function grayImage = rgb2gray2(rgbImage)
try
  [rows, columns, numberOfColorChannels ]= size(rgbImage);

 redChannel = uint16( rgbImage(:, :, 1));
 greenChannel = uint16(rgbImage(:, :, 2));
 blueChannel = uint16(rgbImage(:, :, 3));
 grayImage = 0.2125*double(redChannel) + 0.7159*double(greenChannel) + 0.0721*double
 grayImage = uint8(grayImage);
end
```

2.1.3 RGB-to-GRAY-3

```
function grayImage = rgb2gray3(rgbImage)
try
  [rows, columns, numberOfColorChannels ]= size(rgbImage);

 redChannel = uint16( rgbImage(:, :, 1));
 greenChannel = uint16(  rgbImage(:, :, 2));
 blueChannel = uint16(rgbImage(:, :, 3));
 grayImage = 0.299*double(redChannel) + 0.587*double(greenChannel) + 0.114*double(blueCha
 grayImage = uint8(grayImage);


end
```

2.1.4 Resultat

```
close all
clear all
image=imread('100.jpg');
subplot(2,2,1);
imshow(image);
subplot(2,2,2);
imshow(rgb2gray1(image));
subplot(2,2,3);
imshow(rgb2gray2(image));
subplot(2,2,4);
imshow(rgb2gray3(image));
```




2.2 RGB-to-NGRB

2.2.1 RGB-to-NGRB-1

```
function nrgb1 = rgbtonrgb1(rgbImage)
%(y i q)=( 0.299,0.587,0.114;0.596,-0.274,-0.322;0.211,-0.253,-0.312)*(r g b)
    R =double( rgbImage(:, :, 1));
      G =  double(rgbImage(:, :, 2));
     B = double( rgbImage(:, :, 3));
if R+G+B==0
     r=0;
     g=0;
     b=0;
 else
     r=R./(R+G+B);
     g=G./(R+G+B);
     b=B./(R+G+B);
   rgbImage(:, :, 1)=r*255;
   rgbImage(:, :, 2)=g*255;
   rgbImage(:, :, 3)=b*255;
nrgb1=uint8(rgbImage)
end
```

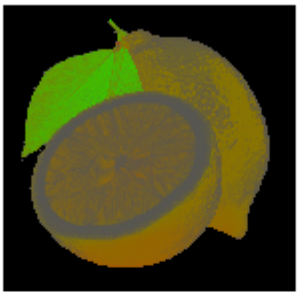
2.2.2 RGB-to-NGRB-2

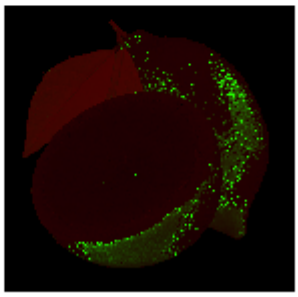
```
function nrgb2 = rgbtonrgb2(rgbImage)
R =double( rgbImage(:, :, 1));
G =double(rgbImage(:, :, 2));
B = double( rgbImage(:, :, 3));
R1=double(R)/255
G1=double(G)/255
B1=double(B)/255
[n,m]=size(R)
H=zeros(n,m)
L=zeros(n,m)
S=zeros(n,m)
for i=1:n
  for j=1:m
   Cmax1=max(R1(i,j),G1(i,j));
   Cmax=max(Cmax1,B1(i,j));
   Cmin1=min(R1(i,j),G1(i,j));
   Cmin=min(Cmin1,B1(i,j));
   thelta=Cmax-Cmin;
   L(i,j)=(Cmax+Cmin)/2;
   if thelta==0
      H(i,j)=0;
   else
    if Cmax==R1(i,j)
    H(i,j)=60*mod((G1(i,j)-B1(i,j))/(thelta),6);
    elseif Cmax==G1(i,j)
    H(i,j)=60*((B1(i,j)-R1(i,j))/(thelta)+2);
    elseif Cmax==B1(i,j)
    H(i,j)=60*((G1(i,j)-B1(i,j))/(thelta)+4) ;
    end
    end
    if thelta==0
    S(i,j)=0;
    else
    S(i,j)=thelta/(1-abs(2*L(i,j)));
    end
    end
```

```
 end
    rgbImage(:, :, 1)=H*255/360;
    rgbImage(:, :, 2)=S*255/100;
    rgbImage(:, :, 3)=L*255/100;
    nrgb2=uint8(rgbImage)
 end
```

2.2.3 Resultat

```
close all
clear all
image=imread('100.jpg');
subplot(2,2,1);
imshow(image);
subplot(2,2,2);
imshow(rgb2gray1(image));
subplot(2,2,3);
imshow(rgb2gray2(image));
subplot(2,2,4);
imshow(rgb2gray3(image));
```





2.3 III

2.3.1 III

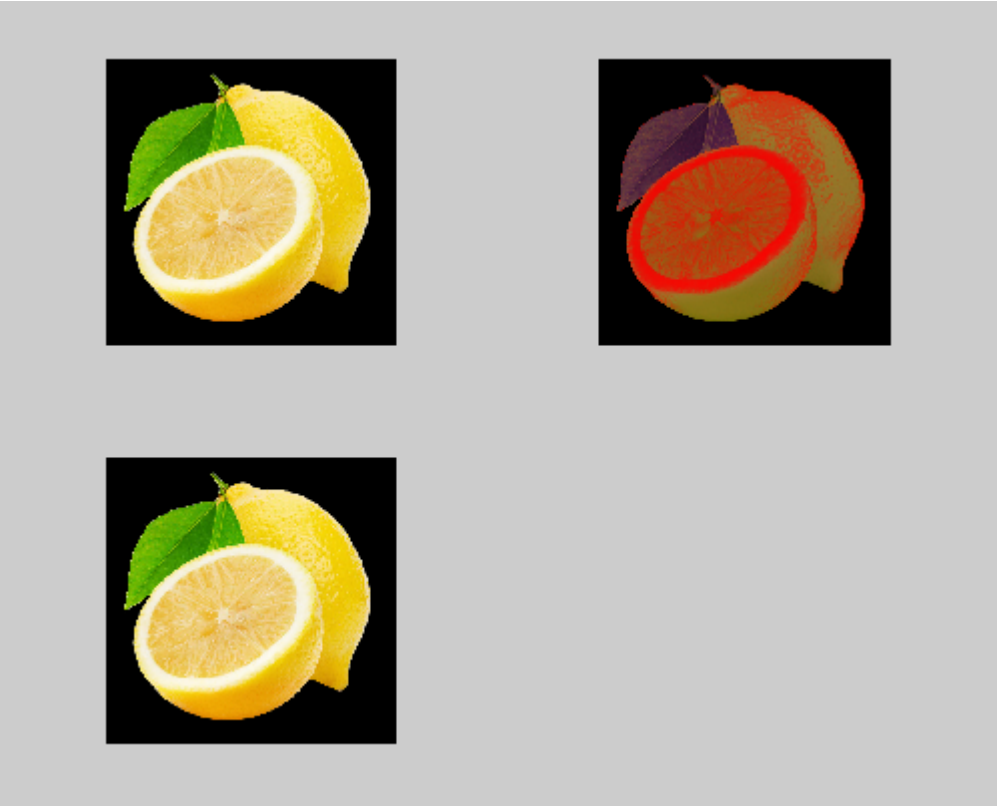
```
function I = III(rgbImage)
redChannel =double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
r=(redChannel+greenChannel+blueChannel)/3;
g=(redChannel-blueChannel)/2;
b=(-1*redChannel+2*greenChannel-blueChannel)/4;
rgbImage(:, :, 1)=r*255;
rgbImage(:, :, 2)=g*255;
rgbImage(:, :, 3)=b*255;
I=uint8(rgbImage)
end
```

2.3.2 III-inverse

```
function I = III_inv(rgbImage)
redChannel =double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
r=redChannel+greenChannel-0.6667*blueChannel;
g=redChannel+1.3333*blueChannel;
b=1*redChannel-1*greenChannel-0.6667*blueChannel;
rgbImage(:, :, 1)=r*255;
rgbImage(:, :, 2)=g*255;
rgbImage(:, :, 3)=b*255;
I=uint8(rgbImage)
end
```

2.3.3 Resultat

```
image2=III(image);
image3=III_inv(image2);
subplot(2,2,1),imshow(image)
subplot(2,2,2),imshow(image2);
subplot(2,2,3),imshow(image3);
```



2.4 RGB/YIQ

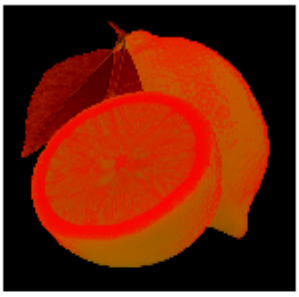
2.4.1 RGB/YIQ

```
function yiq = rgbtoyiq(rgbImage)
%(y i q)=( 0.299,0.587,0.114;0.596,-0.274,-0.322;0.211,-0.253,-0.312)*(r g b)
redChannel =double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
y=0.299*redChannel+0.587*greenChannel+0.114*blueChannel;
i=0.595716*redChannel-0.274453*greenChannel-0.321263*blueChannel;
q=0.211456*redChannel-0.522591*greenChannel+0.311135*blueChannel;
rgbImage(:, :, 1)=y*255;
rgbImage(:, :, 2)=i*255;
rgbImage(:, :, 3)=q*255;
yiq=uint8(rgbImage)
end
```

```
function rgb = yiqtorgb(rgbImage)
redChannel = double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
r=1*redChannel+0.9563*greenChannel+0.6210*blueChannel;
g=1*redChannel-0.2721*greenChannel-0.6474*blueChannel;
b=1*redChannel-1.1070*greenChannel+1.7046*blueChannel;
rgbImage(:, :, 1)=r*255;
rgbImage(:, :, 2)=g*255;
rgbImage(:, :, 3)=b*255;
rgb=uint8(rgbImage)
end
```

2.4.2 Resultat

```
image2=rgbtoyiq(image);
image3=yiqtorgb(image2);
subplot(2,2,1),imshow(image);
subplot(2,2,2),imshow(image2);
subplot(2,2,3),imshow(image3);
```



2.5 RGB/YUV

2.5.1 RGB/YUV

```
function yuv = rgbtoyuv(rgbImage)
%(y i q)=( 0.299,0.587,0.114;0.596,-0.274,-0.322;0.211,-0.253,-0.312)*(r g b)
redChannel =double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
y=0.299*redChannel+0.587*greenChannel+0.114*blueChannel;
u=-0.14713*redChannel-0.28886*greenChannel+0.436*blueChannel;
v=0.615*redChannel-0.51498*greenChannel-0.10001*blueChannel;
rgbImage(:, :, 1)=y*255;
rgbImage(:, :, 2)=u*255;
rgbImage(:, :, 3)=v*255;
yuv=uint8(rgbImage)
end
```

```
function rgb = yuvtorgb(rgbImage)
%(y i q)=( 0.299,0.587,0.114;0.596,-0.274,-0.322;0.211,-0.253,-0.312)*(r g b)
redChannel =double( rgbImage(:, :, 1));
greenChannel =  double(rgbImage(:, :, 2));
blueChannel = double( rgbImage(:, :, 3));
redChannel=redChannel/255;
greenChannel=greenChannel/255;
blueChannel=blueChannel/255;
r=1*(redChannel)+1.13983*(blueChannel);
g=1*(redChannel)-0.39465*(greenChannel)-0.58060*(blueChannel);
b=1*(redChannel)+2.03211*(greenChannel);
rgbImage(:, :, 1)=r*255;
rgbImage(:, :, 2)=g*255;
rgbImage(:, :, 3)=b*255;
rgb=uint8(rgbImage)
end
```

2.5.2 Resultat

```
image2=rgbtoyuv(image);
image3=yuvtorgb(image2);
subplot(2,2,1),imshow(image);
subplot(2,2,2),imshow(image2);
subplot(2,2,3),imshow(image3);
```

Chapter 3

Segmentation

3.1 main

le fichier main contient tout les appels de fonctions en commentaires.

```
Image =imread('100.jpg');
Image1=imread('house.pgm');
im=rgb2gray(Image);%si c'est pas une image gris
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image)
%subplot(2,2,2),imshow(seuil_manuel(im,60));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image)
%subplot(2,2,2),imshow(seuil_median(im));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image)
%subplot(2,2,2),imshow(seuil_moyenne(im));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image)
%subplot(2,2,2),imshow(isodata1(im,30));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image)
%subplot(2,2,2),imshow(otsu(im));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_moyenne(Image1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ù
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_moyenne_constante(Image1,20));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_mediane(Image1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ù
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_mediane_constante(Image1,20));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_bernsen(Image1));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%subplot(2,2,1),imshow(Image1)
%subplot(2,2,2),imshow(seuillage_niblack(Image1));
```

3.2 seuillage-manuel

```
function Image_sortie = seuil_manuel(Image_entree,seuil)

 [n,m]=size(Image_entree);
 Image_sortie=zeros(n,m)
 for i=1:n
     for j=1:m
         if (Image_entree(i,j)<seuil)
             Image_sortie(i,j)=0;
         else
             Image_sortie(i,j)=250;
         end
     end
 end
end
```

3.2.1 Resultat

3.3 seuillage-manuel

```
function Image_sortie = seuil_manuel(Image_entree,seuil)

 [n,m]=size(Image_entree);
 Image_sortie=zeros(n,m)
 for i=1:n
     for j=1:m
         if (Image_entree(i,j)<seuil)
             Image_sortie(i,j)=0;
         else
             Image_sortie(i,j)=250;
         end
     end
 end
end
```

3.3.1 Resultat

3.4 seuillage-mediane

```
function Image_sortie = seuil_median(Image_entree)
  seuil = median(Image_entree(:));
 [n,m]=size(Image_entree);
 Image_sortie=zeros(n,m)
 for i=1:n
     for j=1:m
         if (Image_entree(i,j)<seuil)
             Image_sortie(i,j)=0;
         else
             Image_sortie(i,j)=250;
         end
     end
 end
end
```

3.4.1 Resultat

3.5 seuillage-moyenne

```
function Image_sortie = seuil_moyenne(Image_entree)
  seuil = mean(Image_entree(:));
 [n,m]=size(Image_entree);
 Image_sortie=zeros(n,m)
 for i=1:n
     for j=1:m
         if (Image_entree(i,j)<seuil)
             Image_sortie(i,j)=0;
         else
             Image_sortie(i,j)=250;
         end
     end
 end
end
```

3.5.1 Resultat

3.6 ISODATA

```
function  Image_sortie = isodata1(Image_entree,epsilon)
[n,m]=size(Image_entree);
histogramme=zeros(256,1);
 for i=1:n
     for j=1:m
     histogramme(Image_entree(i,j)+1)=histogramme(Image_entree(i,j)+1)+1;
    end
 end
S(1)=1;
%S(2)=254;
S(2)=round(rand*254)
while S(2)-S(1)>epsilon
Uc1nom=0;
Uc1den=0;
Uc1=0;
Uc2nom=0;
Uc2den=0;
Uc2=0;
 for i=0:S(2)
      Uc1nom=Uc1nom+i*histogramme(i+1);
   end
   for i=0:S(2)
    Uc1den=Uc1den+histogramme(i+1);
   end
    Uc1=Uc1nom/Uc1den;
   for i=S(2)+1:255
        Uc2nom=Uc2nom+i*histogramme(i+1);
   end
   for i=S(2)+1:255
      Uc2den=Uc2den+histogramme(i+1);
   end
    Uc2=Uc2nom/Uc2den;
   S(1)=S(2)
   S(2)=(Uc1+Uc2)/2;
end
Image_sortie = seuil_manuel(Image_entree,S(1));
```

3.6.1 Resultat

3.7 OTSU

```
function Image_sortie = otsu1_1(Image_entree)
[Colonne Ligne] = size(Image_entree);

% Nombre de pixel dans l'image %
NBTotal = Colonne*Ligne ;
% Calcul de l'histogramme %
Histogramme = imhist(Image_entree)';
% Probabilité de chaque nombre de pixel %
Proba= Histogramme/NBTotal;
for i = 1 : 255
     % Calcul de la probabilité des classes %
    Proba1 = Proba(1 : i);
    Proba2 = Proba(i+1 : 256);
    P1 = sum(Proba1);
    P2 = sum(Proba2);
    % Calcul de la moyenne des classes %
    n1 = 0:i-1;
    n2 = i:255;
    Moy1 = sum( n1.*Proba1)/P1;
    Moy2 = sum( n2.*Proba2)/P2;
    % Calcul de la variance des classes %
    Var1 = sum(((n1 - Moy1).^2).*Proba1);
    Var2 = sum(((n2 - Moy2).^2).*Proba2);

    VarianceIntraClasse(i) = Var1 + Var2;
end
[Val,Indice] = min(VarianceIntraClasse(1:255));
level = (Indice-1)/255
Image_sortie = im2bw(Image_entree,level);
Image_sortie = Image_sortie*255;
```

3.7.1 Resultat

Chapter 4

Seuillage

J'ai commencé par réaliser une fonction qui me permets d'extraire une matrice 3 X 3, pour l'appliquer à chaque fois à une des méthodes.

```
function  Sortie = extraire_matrice(matrice,i,j)
Sortie=matrice(i-1:i+1,j-1:j+1)
return
```

4.1 Seuillage-moyen

```
function Image_sortie = seuillage_moyenne(Image_entree)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        moyenne=mean(mean(extraire_matrice(Image_entree,i,j)));

        if Image_entree(i,j)>moyenne
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

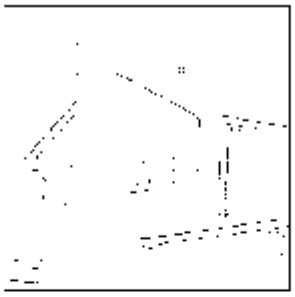
4.1.1 Resultat

4.2 Seuillage-moyenne-constante

```
function Image_sortie = seuillage_moyenne_constante(Image_entree,constante)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        moyenne=mean(mean(extraire_matrice(Image_entree,i,j)));

        if Image_entree(i,j)>moyenne-constante
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

4.2.1 Resultat



4.3 Seuillage-mediane

```
function Image_sortie = seuillage_mediane(Image_entree)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        mediane=median(median(extraire_matrice(Image_entree,i,j)));

        if Image_entree(i,j)>mediane
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

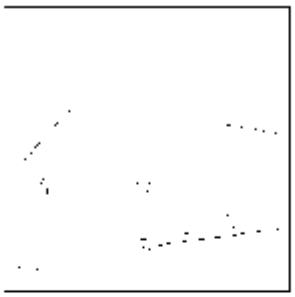
4.3.1 Resultat

4.4 Seuillage-mediane-constante

```
function Image_sortie = seuillage_mediane_constante(Image_entree,constante)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        mediane=median(median(extraire_matrice(Image_entree,i,j)));

        if Image_entree(i,j)>mediane-constante
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

4.4.1 Resultat



4.5 Seuillage-bernsen

```
function seuil = bernsen(Image_entree)
maxi=max(max(Image_entree));
mini=min(min(Image_entree));
seuil=(maxi+mini)/2;
return
```

```
function Image_sortie = seuillage_bernsen(Image_entree)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        seuil=bernsen(extraire_matrice(Image_entree,i,j));

        if Image_entree(i,j)>seuil
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

4.5.1 Resultat

4.6 Seuillage-niblack

```
function seuil=niblack(Image_entree)
moyenne=mean(mean(Image_entree));
ecartype=std(Image_entree(:));
seuil=moyenne+0.2*ecartype;
return
```

```
function Image_sortie = seuillage_niblack(Image_entree)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        seuil=niblack(extraire_matrice(Image_entree,i,j));

        if Image_entree(i,j)>seuil
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

4.6.1 Resultat

4.7 Seuillage-sauvola

```
function seuil=sauvola(Image_entree)
moyenne=mean(mean(Image_entree));
ecartype=std(Image_entree(:));
seuil=moyenne*(1+0.5*((ecartype/128)-1));
return
```

```
function Image_sortie = seuillage_sauvola(Image_entree)
[n,m]=size(Image_entree);
Image_sortie=zeros(n,m);
for i=2:n-1
    for j=2:m-1
        seuil=sauvola(extraire_matrice(Image_entree,i,j))

        if Image_entree(i,j)>seuil
            Image_sortie(i,j)=255;
        else
            Image_sortie(i,j)=0;
        end
    end
end
return
```

4.7.1 Resultat