TP2_Image_MAP201

Groupe: INF2

Binôme: Shaghayagh HAJMOHAMMADKASHI

Kaiwen ZHENG

#Exo5

Code scilab transformations.sci :

```
// chargement des fonctions necessaires au TP
exec("init_tp_image.sce",-1);

// définition de la fonction ramenant les valeurs entre 0 et 255
deff('z = min0_max255(y)','z=min(max(y,0),255)');

////// Histogramme cummulé /////

function Hist=hist_cumul(im)
  // classes pour le calcul de l'histogramme
  classes = [-0.1, linspace(0,255,256)];
  // calcul de l'histogramme
  hist = histc(classes, im, normalization=%f);
  // calcul de l'histogramme cumulé
  Hist = zeros(256);
  //à compléter
  for i = 1:256
    Hist(i) = sum(hist(1:i));
  end
endfunction

////// correction affine ///////
deff("q = f_affine(p, p0, p1)","q =(p-p0)*(255/(p1-p0))");// A MODIFIER
function im2=T_affine(im, p0, p1)
  im2 = f_affine(im, p0, p1);    // appliquer la fonction
  im2 =  min0_max255(im2); // repasser à des valeurs entre 0 et 255
endfunction

function [p0, p1]=calcul_p0p1(im, s)
  // calcul de l'histogramme cumulé normalisé
  H = hist_cumul(im);
  H_norm = H ./ (size(im, 1) * size(im, 2));
  // détermination de p0
  p0 = 0;
  for i = 1:256
    if H_norm(i) >= s/100
      p0 = i - 1;
      break;
    end
  end

  // détermination de p1
```

```
  p1 = 255;
  for i = 256:-1:1
    if H_norm(i) <= (1 - s/100)
      p1 = i - 1;
      break;
    end
  end
endfunction

////// Egalisation d'histogramme ///////
function im_out=hist_egal(im)
  // calcul de l'histogramme cumulé
  H1 = hist_cumul(im);
  factor = 255/(size(im, 1) * size(im, 2)); // normalisation
  // créer une nouvelle image
  im_out = zeros(size(im));
  // appliquer la transformation à chaque pixel
  for i = 1:size(im, 1)
    for j = 1:size(im, 2)
      im_out(i, j) = factor * H1(im(i, j)+1);
    end
  end
  // retourner la nouvelle image
  im_out = min0_max255(im_out);
endfunction

////// seuillage //////

deff("im3 = produit(im1,im2)","im3 = im1.*im2/255");

function im2=seuillage(im, s)
  [M,N] = size(im);
  im2 = zeros(M,N);
  // à compléter
  for i = 1:M
    for j = 1:N
      if im(i, j) >= s
        im2(i, j) = 255;
      end
    end
  end
endfunction
```

Pour tester mon code :

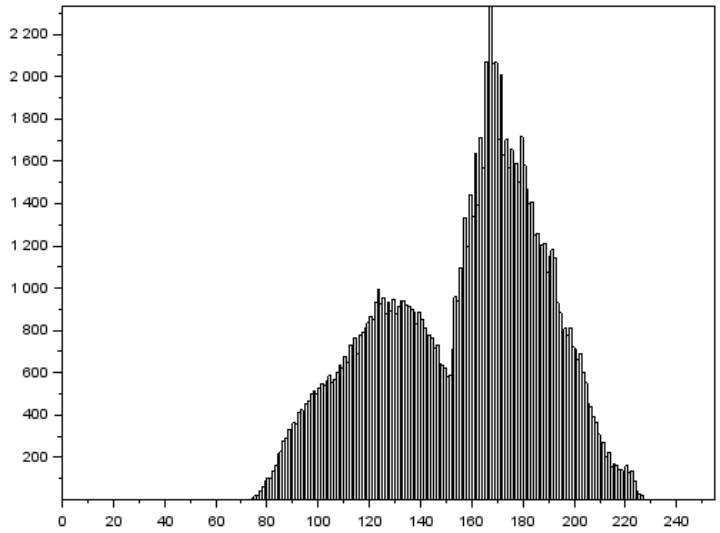
```
exec('transformations.sci');

// desert.bmp
im = lire_imageBMPgris('desert.bmp');
s = 1;
[p0, p1] = calcul_p0p1(im, s);
disp(p0, p1);
im2 = T_affine(im, p0, p1);
im3 = seuillage(im2, p1);
afficher_image(im3);
```

Pour s = 1 et pour image de desert je trouve

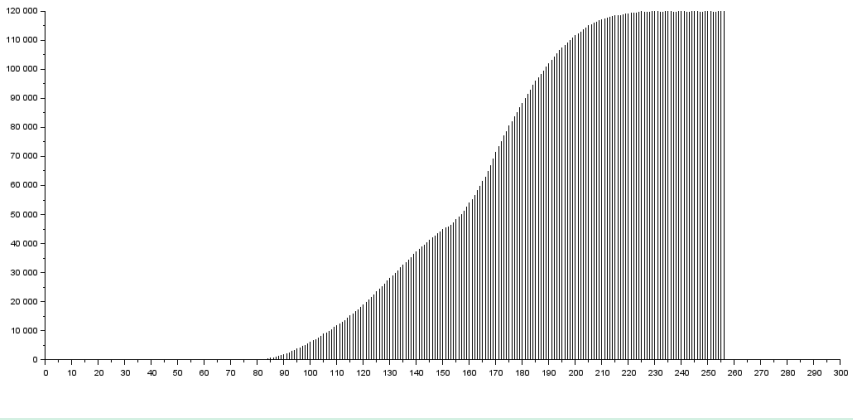
p0= 87

p1=215

Les valeurs de p0 et p1 sont proches de celles que j'ai choisies dans l'exercice précédent p0=74 p1=227



L'histogramme de l'image desert.bmp



L'histogramme cumulé de l'image desert.bmp

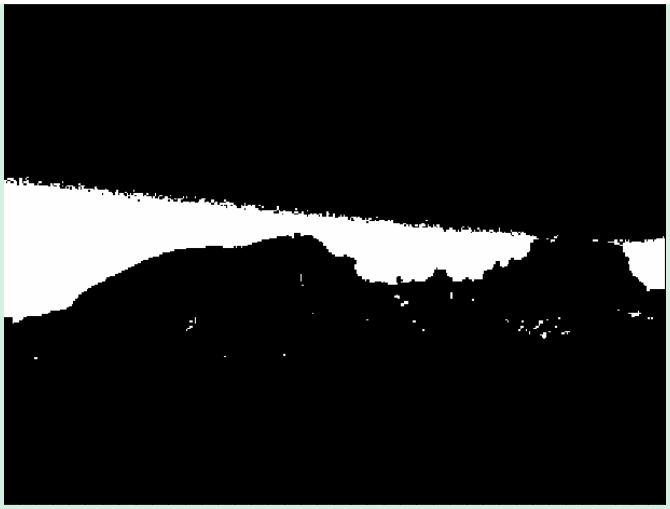


Image desset avec s=1

Avec d’autres valeurs du seuil et affichage des images correspondantes :

Pour s = 10 et pour image de desert je trouve

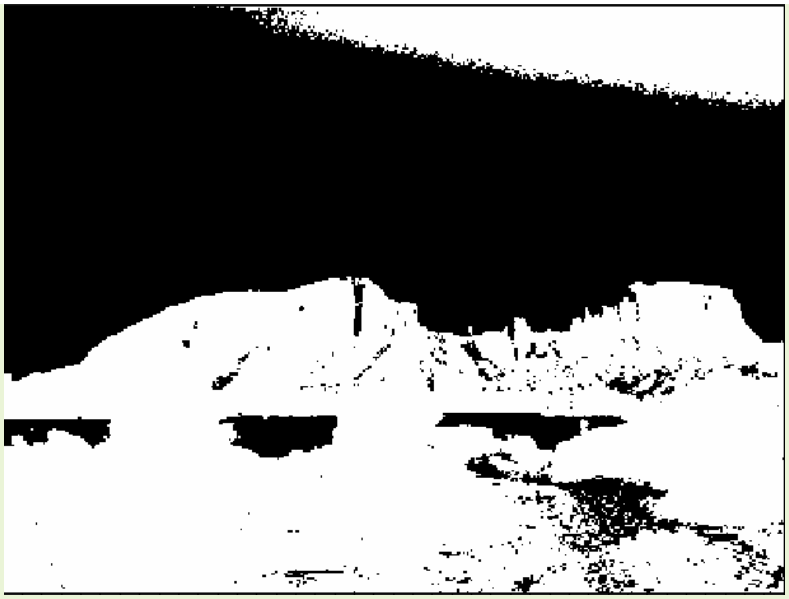
p0= 110

p1=194

Pour s = 50 et pour image de desert je trouve

p0= 164

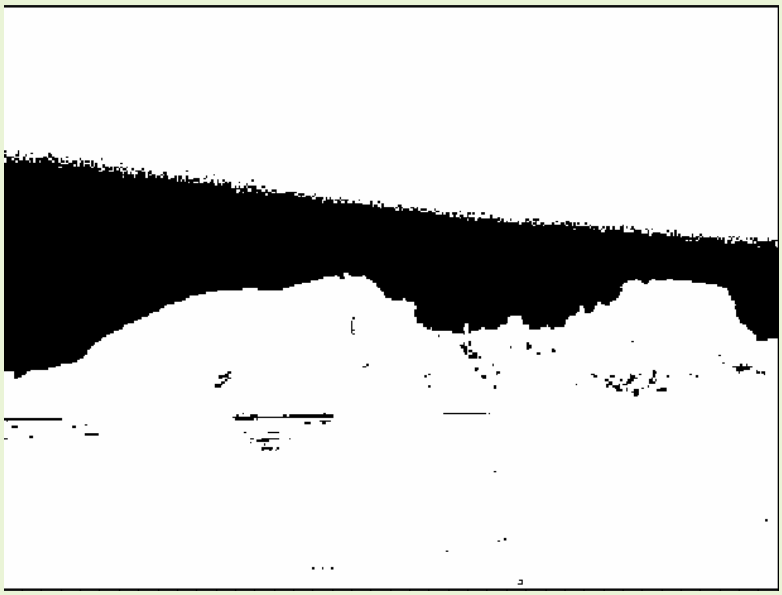
p1=163



Pour s = 100 et pour image de desert je trouve

p0= 230

p1=72



Pour s = 1 et pour image de manoir je trouve

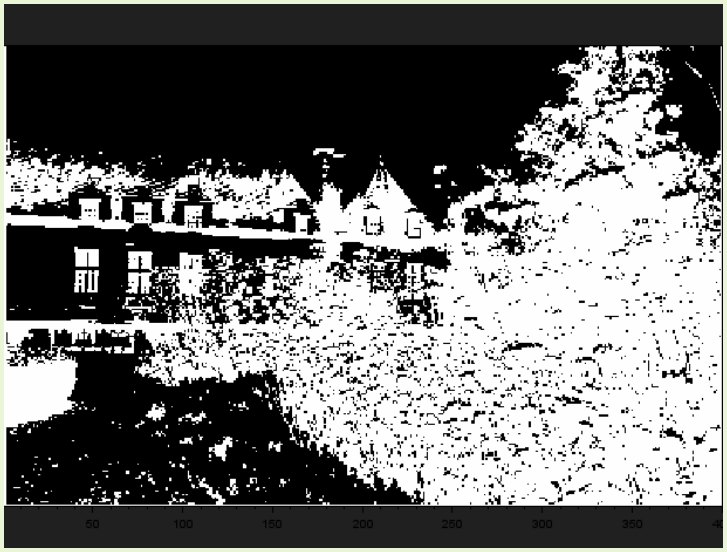
p0= 110

p1=254

Pour s = 50 et pour image de manoir je trouve

p0= 199

p1=198



Pour s = 100 et pour image de manoir je trouve

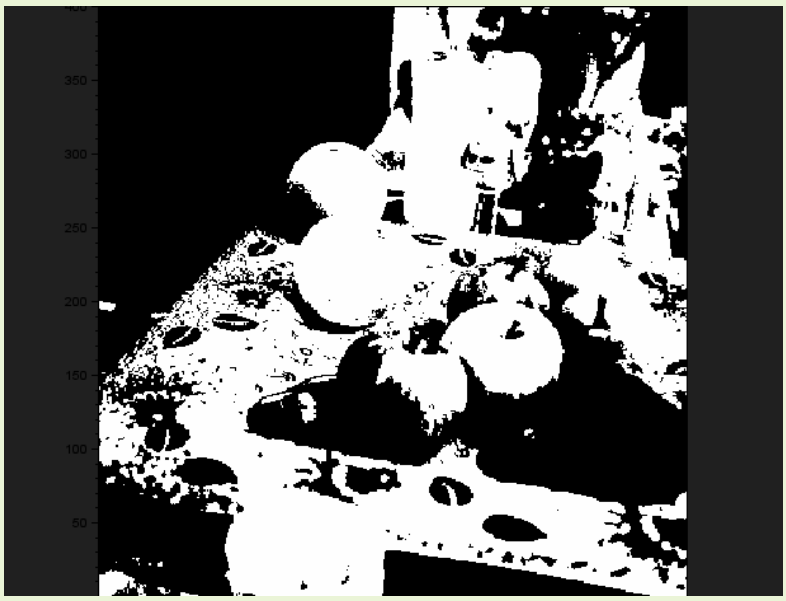
p0= 255

p1=34

Pour s = 1 et pour image de fruits je trouve

p0= 0

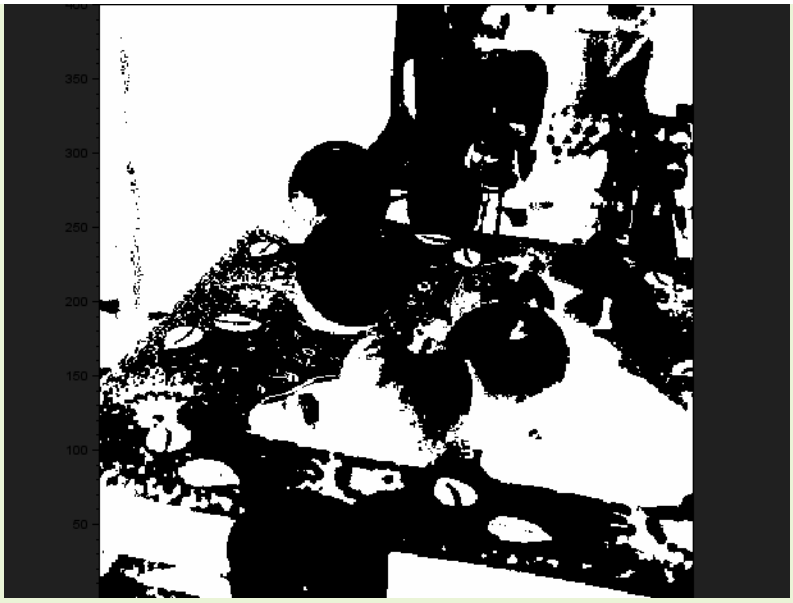
p1=55



Pour s = 50 et pour image de fruits je trouve

p0= 10

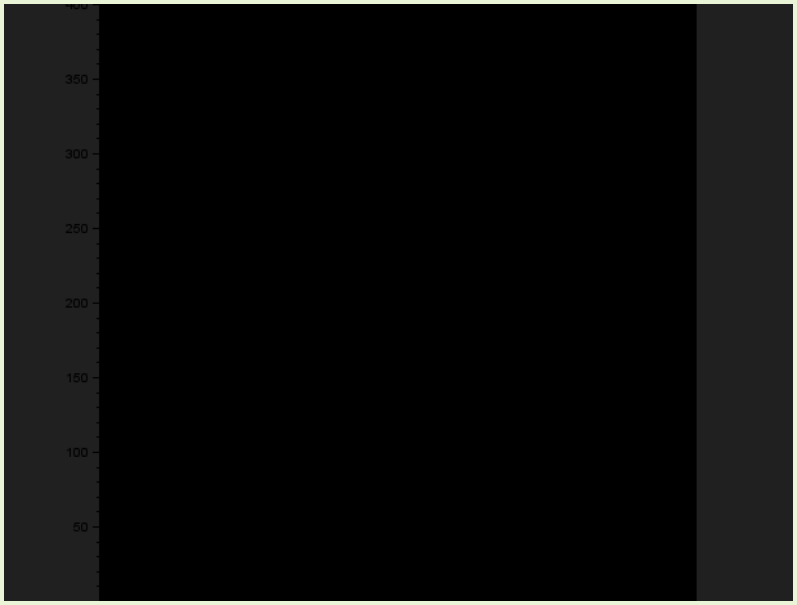
p1=9



Pour s = 100 et pour image de fruits je trouve

p0= 172

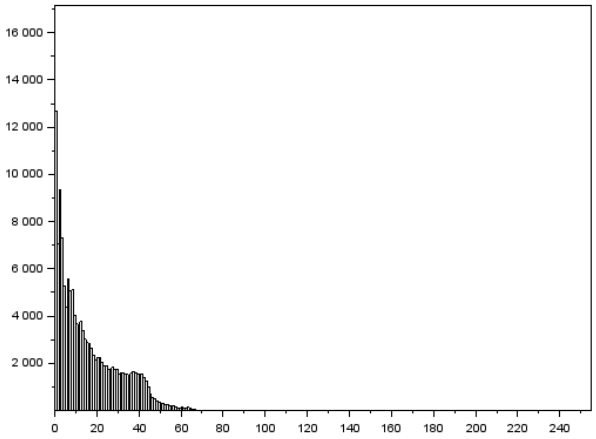
p1=255



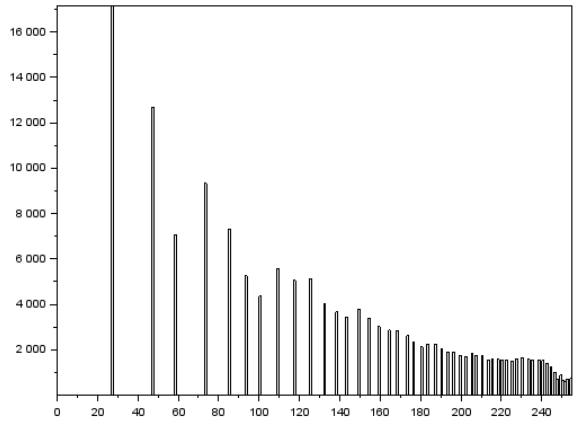
#Exo6

Fruits (Image originale)

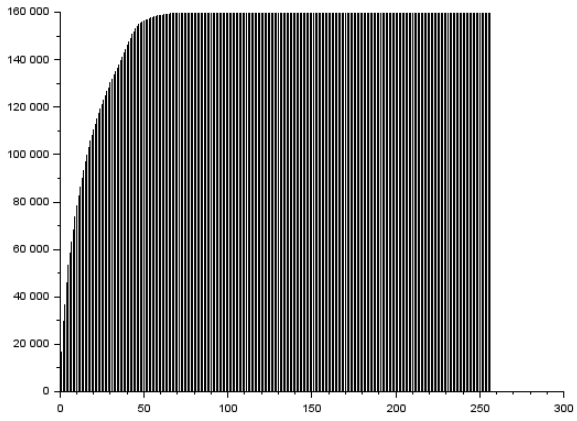
Fruits (Image égalisée)



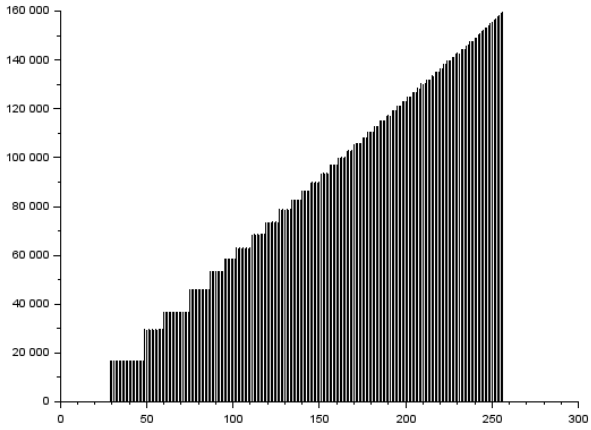
Histogramme de l'image originale " Fruits"



Histogramme de l'image égalisée " Fruits"



Histogramme cumulé de l'image originale " Fruits"



Histogramme cumulé de l'image égalisée " Fruits"

Pour tester mon code :

1: Image égalisée

```
exec('transformations.sci', -1);
im = lire_imageBMPgris('fruits.bmp');
// calcul de l'image égalisée
im_egal = hist_egal(im);
// affichage
afficher_image(im_egal);
```

2: l'histogramme de l'image égalisée

```
exec('init_tp_image.sce');
im = lire_imageBMPgris('fruits.bmp');
// calcul de l'image égalisée
im_egal = hist_egal(im);
// classes pour les valeurs de pixels
classes = [-0.1, linspace(0,255,256)];
// h est un tableau `a 256 entrées qui contient l'histogramme
// h(p) = nb de pixels valant p-1
h = histc(im_egal, classes)
// tracé de l'histogramme
scf();
histplot(classes, im_egal, normalization=%f, strf='021');
```

3: l'histogramme cumulé de l'image égalisée

```
exec('transformations.sci', -1);
im = lire_imageBMPgris('fruits.bmp');
im_egal = hist_egal(im);
// calcul de l'histogramme cumulé
Hist = hist_cumul(im_egal);
// affichage de l'histogramme cumulé
plot2d3(Hist);
```

Pour compléter la fonction "hist_egal" dans le fichier "transformations.sci" pour l'égalisation d'histogramme, on peut utiliser la formule de transformation fournie dans la question :

T(p) = (255/MN)*H1(p)

où H1(p) est l'histogramme cumulé de l'image d'entrée et MN est le nombre total de pixels. Les étapes pour implémenter la fonction sont les suivantes :

- Calculer l'histogramme cumulé de l'image d'entrée en utilisant la fonction "hist_cumul".
- Calculer le facteur de normalisation (255/MN).
- Créer une nouvelle image de même taille que l'image d'entrée.
- Pour chaque pixel de l'image d'entrée, calculer sa nouvelle valeur en utilisant la formule de transformation et l'assigner au pixel correspondant dans la nouvelle image.
- Retourner la nouvelle image.

L'image résultante aura un histogramme plus uniformément distribué, ce qui signifie que le contraste de l'image est amélioré. L'histogramme cumulé de la nouvelle image sera également plus linéaire par rapport à celui de l'image originale.

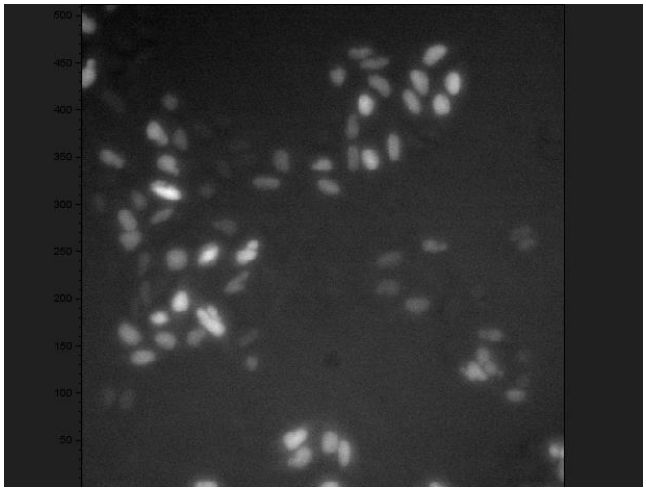
En ce qui concerne l'aspect visuel de la nouvelle image, elle semble plus vive et lumineuse en raison de l'augmentation du contraste. Cependant, il peut y avoir une perte de détails dans les ombres et les hautes lumières.

Par rapport à la correction affine, l'égalisation d'histogramme peut produire des changements plus drastiques sur l'image, car elle redistribue les intensités de pixel sur l'ensemble de la plage. La correction affine, en revanche, ne fait que étirer ou compresser la plage existante des intensités de pixels.

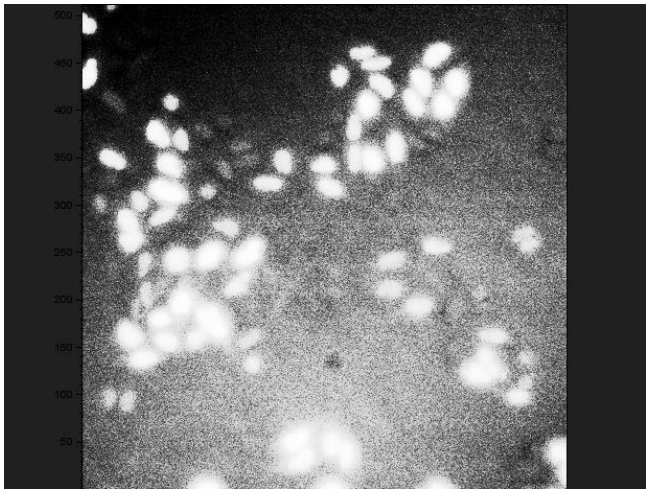
Si nous appliquons à nouveau l'égalisation d'histogramme sur l'image déjà égalisée, nous pouvons observer que l'histogramme devient plus uniforme et que l'image deviens plus lumineuse et les zones sombres deviennent plus sombres. Cela est dû au fait que la formule de transformation augmente encore les intensités de pixel des régions plus lumineuses.

Fruits (Image égalisée après 2 fois)

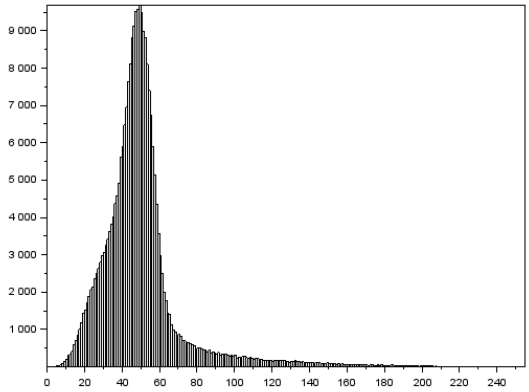
La photo des cellules présente une faible variation de tonalité de couleur, ce qui rend la distribution des intensités de pixels relativement uniforme. Dans ce cas, l'histogramme de l'image est déjà équilibré de manière naturelle, et l'application de la fonction "hist_egal" ne produira pas de changements significatifs dans la qualité visuelle de l'image. Cela peut être dû à plusieurs facteurs tels que l'éclairage uniforme, la nature homogène des cellules ou le type de coloration utilisé pour l'image.



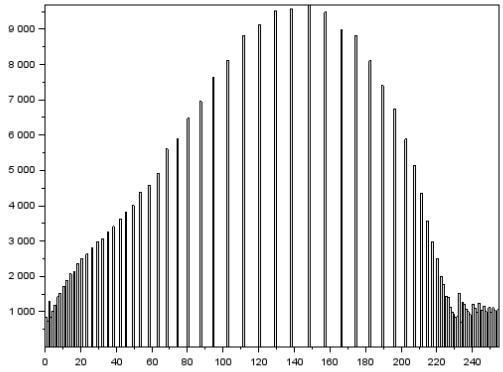
Cellules (Image originale)



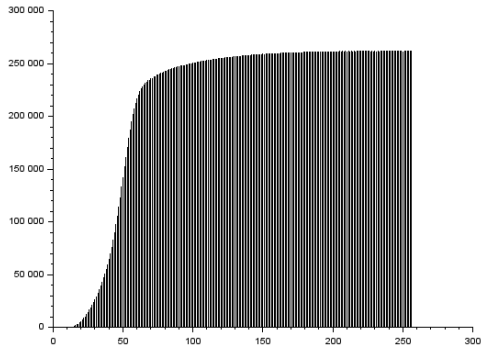
Cellules (Image égalisée)



Histogramme de l'image originale " Cellules"



Histogramme de l'image égalisée " Cellules"



Histogramme cumulé de l'image originale " Cellules"

Histogramme cumulé de l'image égalisée " Cellules"