Compte rendu TME semaine 1 BIMA

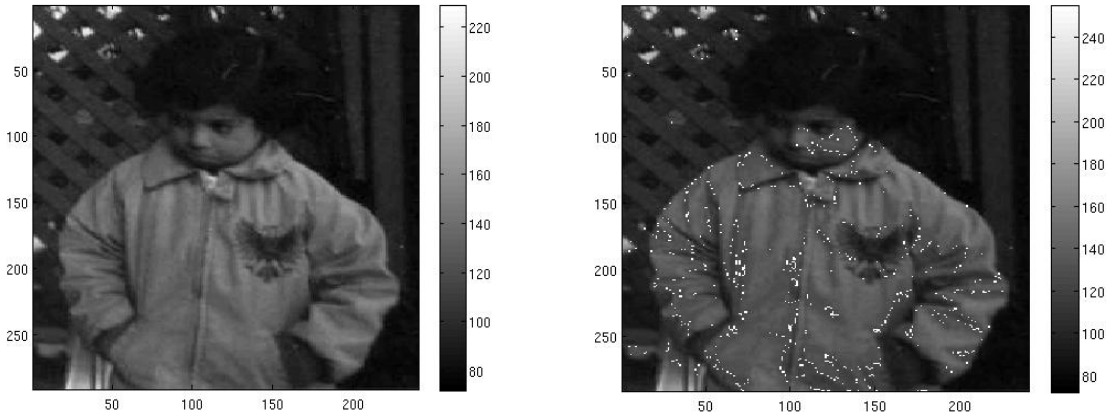
Binôme : Daniel Antunes Costa Gonçalves, Matthieu Wolfrom
Groupe 1

Exercice 1 :

Q1.
Tout d'abord on utilise la fonction imread() pour ouvrir l'image. On utilise ensuite la fonction size() pour obtenir le nombre de lignes et le nombre de colonnes de la matrice de l'image.

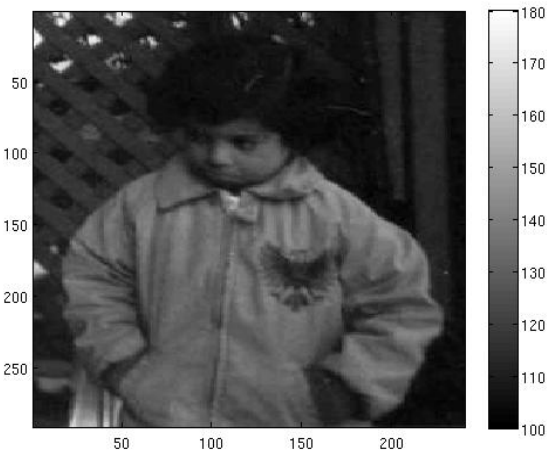
Q2.
Pour compter le nombre de pixels de niveau de gris k d'une image, on parcourt la matrice en vérifiant le niveau de gris des pixels et on incrémente le compteur si le niveau est égal à k.

Q3.
On commence par copier la matrice de l'image originale dans une nouvelle matrice puis on utilise la fonction find() pour remplacer les pixels dont le niveau de gris est k1 par k2.



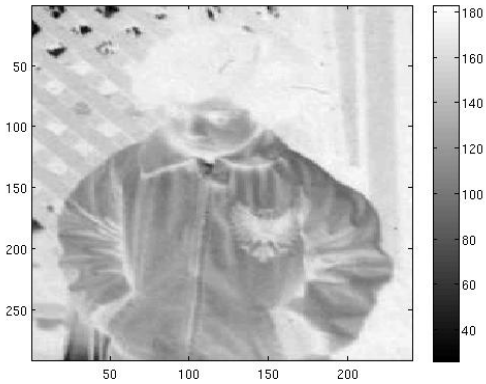
À gauche l'image originale, à droite l'image après remplacement des pixels de niveau de gris k1=125 par des pixels de niveau de gris k2=255. On remarque clairement un grand nombre de pixels blanc dans l'image de droite qui ont remplacé des pixels gris à gauche.

Q4.
Pour normaliser l'image, on cherche à répartir les niveaux de gris dans un nouvel intervalle [k1,k2] plutôt que l'intervalle de départ [Kmin, Kmax].
On parcourt la matrice de l'image et on créé la matrice de l'image normalisée.
Pour cela, on calcule le rapport noté n entre le nouvel intervalle et l'original. On réalise ensuite le produit de n avec la différence entre le niveau de gris du pixel courant et le minimum de l'intervalle original puis on ajoute le minimum du nouvel intervalle pour obtenir le niveau de gris normalisé. L'exemple est ici avec k1 = 100 et k2 = 180.

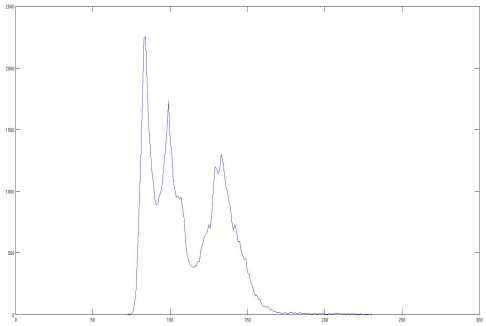


Q5.
Pour créer l'image inversée, il suffit de parcourir la matrice et de construire une matrice inversée avec k' = 255 – k, avec k' le niveau de gris inversé et k le niveau de gris trouvé dans la matrice originale.

Ici on voit l'image avec les niveaux de gris inversés.

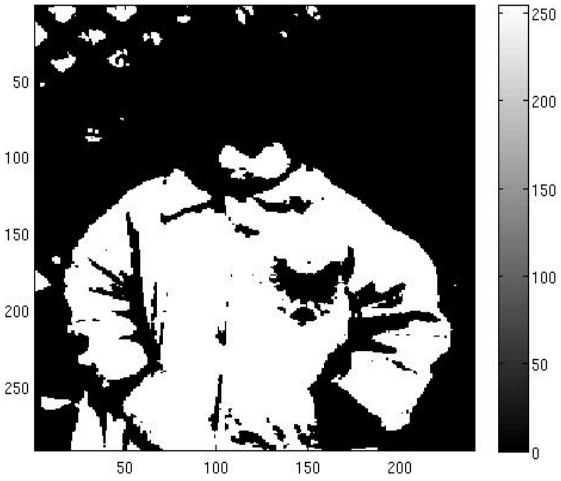


Q6.
Pour obtenir l'histogramme de l'image donnée, on compte les occurrences de chaque niveau de gris de sa matrice, et on les conserve dans un vecteur colonne de taille 256, car il y a 256 niveaux de gris possibles ici.



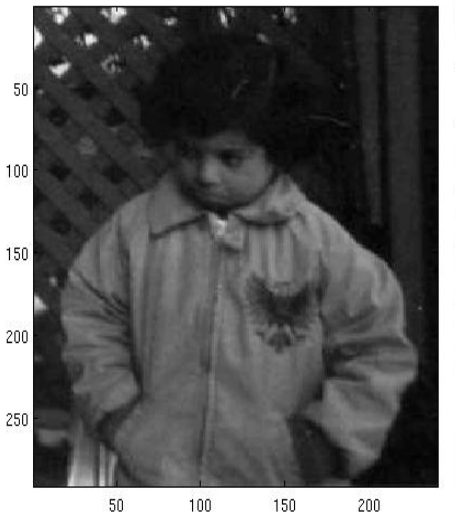
Q7.
Pour construire la matrice de l'image seuillée, on utilise la fonction find() pour trouver les valeurs supérieures au seuil s et les remplacer par 255, puis un second appel à find() permet de mettre à 0 les pixels de valeur inférieure ou égale à s.

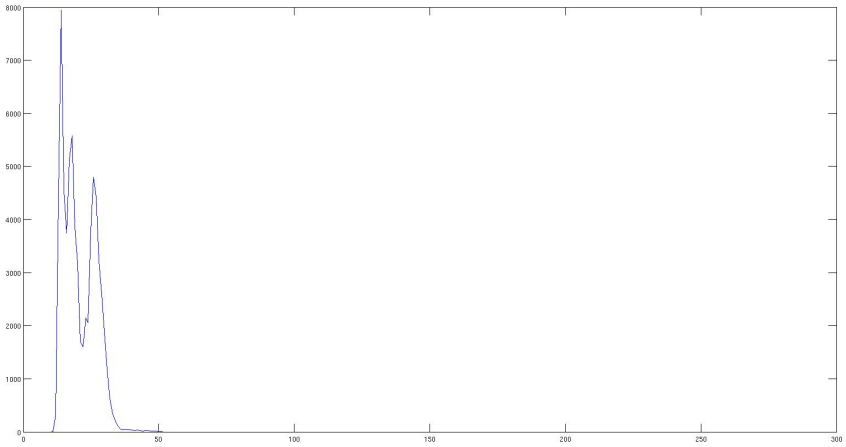
Voici l'image résultante pour un seuil s=120.
Ceci nous permet de bien voir les contours de l'image, ici entre la veste de la fille et le fond de l'image.



Q8.
On commence par ouvrir l'image avec la fonction imread() et on conserve le résultat dans la matrice I. On utilise ensuite la fonction image() pour afficher l'image correspondant à I.
A l'aide de calculerHisto() et plot(), on obtient puis on affiche l'histogramme de I.
Puis, on utilise inversionImage() et calculerHisto() pour obtenir la matrice inversée In ainsi que son histogramme, enfin on affiche ce nouvel histogramme à l'aide de plot() et l'image correspondant à In avec image().

Q9.
On ouvre et on affiche l'image ainsi que son histogramme de la même manière que dans la question précedente.
Puis on utilise normaliseImage() avec k1=10 et k2=50 pour obtenir la nouvelle matrice, dont on calcule l'histogramme avec calculerHisto().
On affiche en suite l'image normalisée avec image() et l'histogramme avec plot().
Voici un exemple d'exécution de cette série d'instructions :





Q10.
La série d'instructions est identique aux deux questions précédentes la seule différence est que l'on appelle seuillerImage() avec s = 128 plutôt que normaliseImage() ou inversionImage().

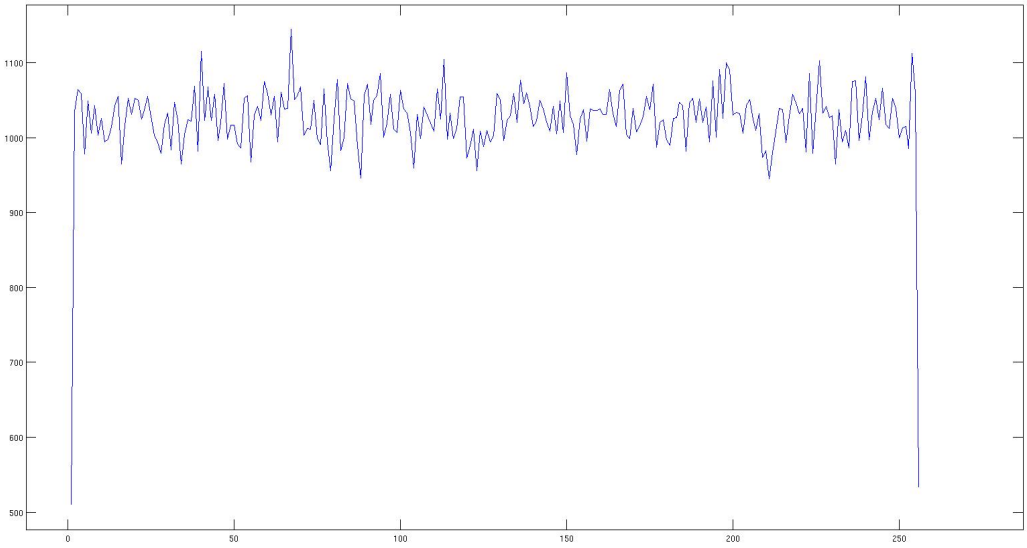
Exercice 2 :

Q1.

128	128	0	255
128	0	0	255
0	128	0	255
128	128	0	255

L'histogramme de cette image est simplement composé de 3 pics en 0, 128 et 255.

Q2.

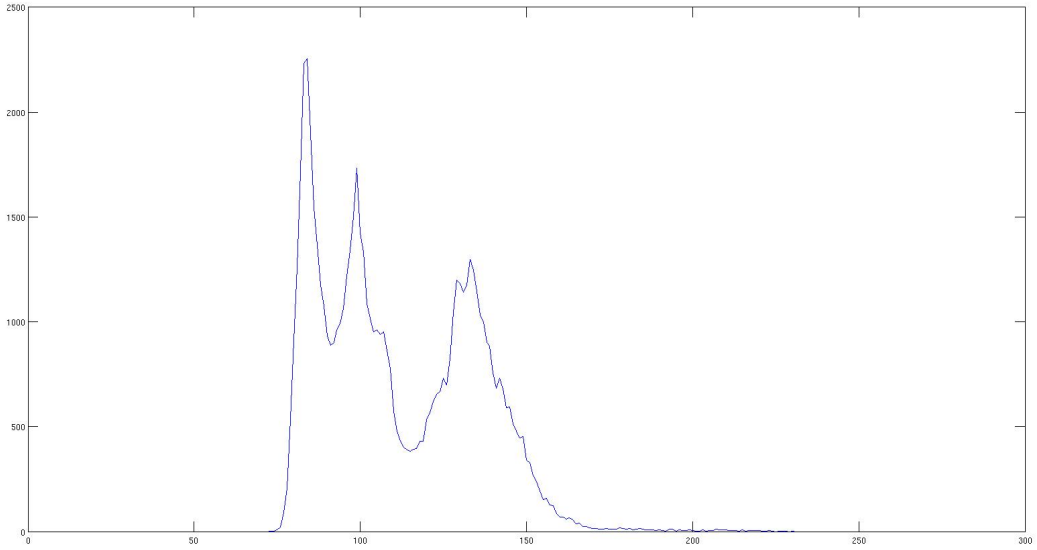


L'histogramme obtenu en générant une matrice de taille 512x512 avec la fonction rand() obtient une distribution très régulière sauf pour les points 0 et 255 où l'on observe des quantités très faibles.

Exercice 3 :

Q1.
L'ouverture de l'image 'pout.tif' sous Matlab ne fonctionnait pas, nous avons donc récupérer l'image suivante sur internet, qui est supposément l'image originale :

Q2.
Voici l'histogramme de cette image :



Q3.
Dans un premier temps on calcule l'histogramme cumulé en utilisant l'histogramme obtenu avec la fonction calculerHisto(), puis on parcourt le vecteur de l'histogramme et on construit le vecteur de l'histogramme cumulé en additionnant progressivement les valeurs de l'histogramme.