Travaux Pratiques 3

TP : Traitement d'images avec Numpy et OpenCV

Objectifs du TP
- Implémenter un filtre de convolution en Python à l'aide de Numpy.
- Calculer l'histogramme d'une image en niveaux de gris et réaliser une égalisation de l'histogramme.
- Appliquer un seuillage simple pour obtenir une image binaire.

Partie 1 : Filtre de convolution avec Numpy

Objectif
Implémenter une fonction Python qui applique un filtre de convolution (kernel) à une image en niveaux de gris.

Consignes
1. Charger une image en niveaux de gris à l'aide de OpenCV.
2. Implémenter une fonction `def convolution(image, kernel):` qui réalise la convolution sans utiliser de fonction prédéfinie (comme cv2.filter2D).
  - Parcourir l'image pixel par pixel.
  - Pour chaque pixel, appliquer la multiplication élément par élément avec le kernel centré sur ce pixel.
3. Tester votre fonction avec différents filtres, par exemple :
  - Un filtre flou (moyenneur) :
   [[1/9, 1/9, 1/9],
    [1/9, 1/9, 1/9],
    [1/9, 1/9, 1/9]]
  - Un filtre de détection de contours (exemple : Sobel horizontal ou vertical).

Comparer votre implémentation avec la fonction d'opencv cv2.filter2D.

Conseils:
- Pour simplifier l'implémentation, vous pouvez ajouter un padding (bordure) autour de l'image.
- Assurez-vous que le résultat a la même taille que l'image d'entrée.

Partie 2 : Calcul et égalisation de l'histogramme

Objectif
Calculer l'histogramme d'une image et réaliser une égalisation de l'histogramme pour améliorer le contraste.

Consignes:
1. Charger une image en niveaux de gris.

2. Calculez son histogramme (compte le nombre de pixels pour chaque intensité de 0 à 255).
3. Implémentez l'égalisation d'histogramme en suivant ces étapes :
  - Calculer la fonction de répartition (cumulative sum) de l'histogramme normalisé.
  - Utiliser cette fonction pour transformer l'image initiale de manière à répartir plus uniformément les intensités.
4. Affichez l'image originale, son histogramme, l'image après égalisation et son histogramme.

Conseils
- Utilisez Matplotlib pour tracer les images et les histogrammes.
- Vérifiez l'effet de l'égalisation sur une image avec un contraste faible (par exemple, une image sous-exposée).

Partie 3 : Seuillage et image binaire

Objectif
Appliquer un seuillage simple à une image en niveaux de gris pour obtenir une image binaire.

Consignes
1. Charger une image en niveaux de gris (peut-être celle après égalisation ou l'image originale).
2. Implémentez une fonction `def seuillage(image, seuil):` qui transforme chaque pixel en 0 si l'intensité est inférieure au seuil et en 255 si elle est supérieure ou égale au seuil.
3. Tester votre fonction avec différentes valeurs de seuil et afficher le résultat (image binaire).

Conseils
- Vous pouvez ajouter une option pour inverser le seuillage (optionnel).
- Affichez l'image binaire ainsi que l'histogramme pour observer l'impact du seuil.

Bonus (optionnel)
2. Comparer les résultats de votre implémentation de l'égalisation d'histogramme avec celle proposée par OpenCV (`cv2.equalizeHist`).