Traitement d'images

Dans ce TP, nous allons voit comment implémenter des algorithmes basiques sur des images.

Nous utiliserons les bibliothèques `numpy`, `matplotlib.pyplot`, `matplotlib.image` et `random`.

Pour importer une image en Python, on utilisera la commande `imread` du module `matplotlib.image`, en donnant le chemin complet vers le fichier voulu (format png).

Une image (en niveaux de gris) sera alors représentée par une matrice de nombres entre 0 et 1.

Une image (en couleur) sera représentée par une matrcice dont les coefficients sont des listes de trois nombres entre 0 et 1 : la quantité de rouge, de vert et de bleu.

On utilisera les images données sur le site : Escher pour les images en noir et blanc, et Lenna pour les images couleurs.

Exercice 1. On peut bruiter une image en ajoutant à chaque pixel un nombre aléatoire. Implémenter cette opération.

Exercice 2. Écrire des fonctions permettant de faire une symétrie horizontale ou verticale à l'image.

Exercice 3. On peut ajuster la luminosité d'une image en ajoutant à chaque pixel un réel de $[0,1]$ choisi. Écrire une fonction permettant d'augmenter ou de diminuer la luminosité.

Exercice 4. On peut pixeliser une image de la façon suivante : si $p \in \mathbb{N}^*$, on découpe l'image en carrés de taille $p \times p$, puis dans chaque carré, on remplace la valeur du pixel par la valeur moyenne dans le carré.

Écrire une fonction implémentant cette opération.

Exercice 5. On peut flouter une en remplaçant la valeur d'un pixel par la valeur moyenne de tous les pixels qui l'entourent (*i.e.* de tous les pixels de la matrice 3×3 centrée en le pixel voulu).

Écrire une fonction implémentant cette opération.

Exercice 6. Plus généralement, on peut appliquer un *produit de convolution* à une image.

Étant donnée une matrice $A = (A_{ij}) \in \mathcal{M}_{a,b}(\mathbb{R})$, on fait pour chaque pixel (i,j) les opérations suivantes :

- on extrait la sous-matrice de taille $a \times b$ centrée en (i,j) de l'image
- on fait le produit terme à terme (attention, pas le produit matriciel) de cette sous-matrice avec A
- on remplace la valeur de (i,j) par la somme de tous les coefficients de la matrice calculée à l'étape précédente

Implémenter cette opération. On pourra tester avec les matrices suivantes :

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \frac{1}{16}\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Exercice 7. Passons maintenant aux images en couleurs. Pour passer d'une image couleur à une image en niveaux de gris, on transforme chaque coefficient de la matrice en un flottant qui est une moyenne pondérée : gris = $0.299\times$ rouge + $0.587\times$ vert + $0.114\times$ bleu.

Écrire une fonction calculant l'image en niveaux de gris d'une image couleurs.

Exercice 8. Réimplémenter les fonctions changeant la luminosité, pixellisant une image, et faisant une produit de convolution pour les images couleurs.

Exercice 9. On va dans cette question définir le filtre médian. Pour chaque pixel, on le remplace par la médiane des neufs valeurs de la matrice 3×3 centrée en ce pixel.

Pour tester cette fonction, on pourra implémenter le script suivant :

- On bruite l'image, en changeant certains pixels par des pixels blancs (par exemple, avec probabilité 0.1)
- On applique le filtre médian, éventuellement plusieurs fois, jusqu'à faire disparaître le bruit.