

Le traitement des images

Le programme d'informatique limite le traitement des images en niveau de gris, ce qui permet de travailler avec une matrice à deux dimensions.

Ceci étant, il est d'étendre les différentes techniques à des images en couleurs en les applications successivement aux matrices R, G et B, voire A (correspondant à la transparence)

Le format de travail est le format png. On utilise numpy pour traiter les matrices, matplotlib.pyplot pour les afficher.

Exercices

Ecrire le programme en Python répondant aux questions suivantes

Exercice 1 : Obtenir une image en niveau de gris à partir d'une image en couleur.

Principe : Pour avoir des niveau de gris, on ne retient que la matrice d'une des couleurs R G ou B au choix ou on calcule une matrice qui serait la moyenne des matrices R, G et B.

Exercice 2 : Négatif : on obtient le négatif d'une photo en remplaçant les noirs par les blancs et réciproquement

Exercice 3 : Binarisation de l'image : En traitement d'image, c'est une opération qui produit des pixels noirs ou blanc.

Principe : Pour cela, on définit un seuil : au-delà, c'est blanc (1), en deçà, c'est noir (0)

Exercice 4 : Réduction par un facteur 3

Principe : on remplace un carré 3x3 par une cellule contenant la moyenne

Exercice 5 : Pixellisation

Principe : L'image est divisée en rectangles de la taille spécifiée taille*taille. Chaque rectangle est ensuite rempli avec la couleur moyenne de la zone.

Exercice 6 : Symétrie inverser une image, faire une symétrie horizontale, verticale.

Exercice 7 : Éclaircir

Principe : on augmente chaque pixel de x ou on multiple chaque pixel par x. Attention à ne pas dépasser 1

Exercice 8 : Flou

Principe : on rend l'image flou en remplaçant la valeur de chaque pixel par la moyenne des 9 pixels formant un carré.

Exercice 9 : Accentuation

Principe : on rend l'image plus nette en remplaçant un pixel par la moyenne coefficientée des points qui l'entourent en suivant la matrice $\begin{bmatrix} 0, & -0.5, & 0 \\ -0.5, & 2, & -0.5 \\ 0, & -0.5, & 0 \end{bmatrix}$

Exercice 10 : Photos maton

Principe : on sort une plaque de nxn photos réduites de l'original. (prendre n=3).

Exercice 11 : Extraction, on extrait une partie de la photo

Indications :

- La structure globale d'une fonction est

```
def Nom_fonction{img,xxx):  
    ...  
    return img
```

On utilise ensuite cette fonction dans le cadre (Nom est une variable contenant le nom de l'image que l'on souhaite traitée).

```
import numpy as np  
import matplotlib.pyplot as plt  
Image = plt.imread(nom+'.png')  
Image = Nom_fonction(Image,xxx)  
plt.imshow(img,cmap=cm.gray) ; plt.show()
```

- Avec numpy toute opération sur les matrices correspondent à des opérations coefficient par coefficient, donc il est possible pour certaine fonction de ne faire qu'une opération matricielle.

- sinon, l'algorithme consiste à créer un tableau img2 de mêmes dimensions (sauf exception) que img formé de 0, puis à le parcourir pour le modifier cellule par cellule.

```
def Nom_fonction{img,xxx):  
    l,c=img.shape  
    img2=np.zeros([l,c])  
    for i in range(l):  
        for j in range(c):  
            img2[i,j]=...  
    ...  
    return img
```