Traitement d'images en Python

Une image est un ensemble de pixels qu'on peut manipuler en python sous forme d'un tableau numpy. Lorsque l'image est en niveaux de gris, le tableau est bidimensionnel et contient des entiers entre 0 (Noir) et 255 (Blanc) donnant ainsi 256 intensités de niveaux de gris du pixel de coordonnées (i,j) ou i est l'indice ligne et j l'indice colonne. Cependant, les niveaux de gris peuvent être représentés par des valeurs réelles de 0 (Noir) à 1 (Noir), ceci dépend du format de l'image ainsi que du module utilisé.

Lorsque l'image est en couleur, le tableau est à 3 dimensions. La troisième dimension contient une entrée à 3 valeurs [r, g, b] représentant le codage de la couleur (l'intensité des 3 couleurs : rouge, vert, bleu). Ce codage s'appelle RGB pour Red Green Blue.

Manipulation d'une image couleur :

1. Ecrire une fonction python qui permet de charger l'image 'Penguins.png' qui se trouve dans le dossier pyzo2015a et de retourner un tableau numpy T représentant cette image.
   N.B : on donne la fonction *imread* du sous module image de *matplotlib* qui permet de lire une image donnée en paramètre sous forme de fichier (il faut préciser le chemin d'accès de l'image si ce n'est pas sous pyzo2015a sinon juste mentionner le nom et l'extension du fichier) et retourne un tableau numpy représentant l'image.
2. Testez votre fonction en affichant le type, la dimension du tableau T ainsi que le dtype des éléments du tableau.
3. Ecrire une fonction python qui permet d'afficher à l'écran une image donnée en paramètre.
   N.B : On donne les fonctions suivantes du sous module pyplot de matplotlib :
   - La fonction *imshow* permet de charger en mémoire une image donnée sous forme d'un tableau numpy
   - La fonction *show* permet l'affichage à l'écran.
4. Ecrire trois fonctions qui permettent de retourner un tableau numpy représentant l'image donnée en paramètre en considérant à chaque fois une seule composante de couleur (bleue, vert et rouge)
5. Ecrire une fonction qui permet d'afficher quatre images dans un même graphique en associant à chaque image une étiquette avec xlabel ('étiquette'). Testez votre fonction en affichant l'image d'origine et celles de la composante bleue, verte et rouge.
   N.B : On donne la fonction subplot du sous module pyplot de matplotlib qui permet d'afficher plusieurs figures sur un même graphique. La syntaxe de

subplot est la suivante : sbplot (L, C, N) ou L est le nombre de lignes, C le nombre de colonnes et N un compteur.

6. Ecrire une fonction python qui permet de retourner une image en niveaux de gris à partir d'une image couleur(donnée sous forme de tableau numpy) sachant que le niveau de gris d'un pixel de l'image retournée est la moyenne des valeurs des composantes RGB de l'image couleur .

Manipulation d'une image en niveaux de gris :

Dans toutes les fonctions qui vont suivre, nous allons travailler avec une image en niveaux de gris. On considérera l'image résultat de la fonction de la question 6.

1. Ecrire une fonction python qui permet de construire le négatif d'une image en niveaux de gris passée en paramètre. Le négatif d'une image est obtenu en inversant l'échelle de niveaux de gris. Vous pouvez définir une fonction lambda qui retourne l'inverse d'un niveau de gris.
2. Ecrire deux fonctions python qui permettent de construire respectivement l'image symétrique d'une image par rapport à l'axe vertical et celle symétrique par rapport à l'axe horizontal (effet miroir)
3. Pour améliorer le contraste, on peut transformer le niveau de gris d'entrée en un nouveau niveau dans le but d'écarter les différences entre les niveaux. Pour cela, on applique une fonction bien choisie. Définir une fonction python qui permet d'appliquer les deux fonctions suivantes sur une image en niveaux de gris (Testez et interprétez) :
   - $f(x)= \sqrt{x}$
   - $g(x)= \frac{1}{2} + \frac{1}{2} \times \sin(\pi (x - \frac{1}{2}))$

N.B : définir f et g comme deux fonctions lambda.

4. La convolution est une technique de traitement d'image ou la valeur de chaque pixel est recalculée en fonction des valeurs des pixels voisins en utilisant un filtre (masque) de convolution (généralement assez petit d'ordre impair). L'opération consiste à balayer l'image par le filtre, et à affecter au pixel de coordonnées (x, y) la moyenne des pixels voisins pondérés par les coefficients du filtre (on ignore les pixels de bords).

On considérera qu'un filtre est une liste de listes représentant une matrice 3 × 3.

Exemple de filtres :

F1 = [[1, 0, -1], [0, 0, 0], [-1, 0, 1]]

F2 = [[0, 1, 0], [1, -4, 1], [0, 1, 0]]

F3 = [[0, 1, 0], [1, 1, 1], [0, 1, 0]]

a) Ecrire une fonction python Convolation_pix (I, F, x, y) qui permet de retourner la nouvelle valeur du pixel de coordonnées (x, y) d'une image I en considérant le filtre F.
b) Ecrire une fonction python Convolution (I, F) qui permet de retourner une matrice de la même taille que I résultat de la convolution de I par le filtre F.
c) Testez la convolution avec les filtres F1, F2 et F3. Interprétez.