

Traitement d'images

TP N°1 : Initiation à la manipulation d'images numériques

Avant de commencer, veuillez installer **Spyder (Python 3.6)** et la bibliothèque **OpenCV 4.2.0** ou vous pouvez travailler sur Google Colab

Importer les modules suivants :

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import cmath
import os
```

Manipulation 1

Soient les vecteurs suivants : $A=[0,4,8,12, \dots,252]$, $B=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ et $C=[0\ 0\ 1\ 1\ 1\ 1\ 0\ 0]$.

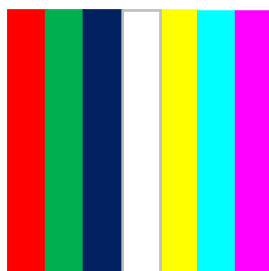
- 1- Créer les matrices suivantes :
 - a. I1 ayant 64 lignes identiques au vecteur A
 - b. I2 constitué d'un bloc de 4X4 de '1' entouré de deux rangs de 0 (à partir des vecteur B et C)
 - c. I3 une matrice carrée obtenue par une reproduction 2x2 de la matrice I2.
- 2- Convertir les 3 matrices en uint8 à travers l'instruction `np.uint8(nom de la matrice)` et visualiser les à l'aide des deux manières suivantes . Commenter.

```
cv.imshow("Image", nom de la matrice)
cv.waitKey(0)
cv.destroyAllWindows()
```

```
plt.imshow(nom de la matrice, cmap = 'gray')
plt.show()
```

Manipulation 2

Créer et afficher une image couleur de taille 28x28 pixels telle que représentée sur la figure ci-dessous en utilisant l'instruction `cv.merge([.,.,.,.])`



Traitement d'images

1. Sur combien de bits est codée l'image obtenue ?
2. Calculer son poids théorique
3. Enregistrer l'image obtenue au format .bmp à l'aide de l'instruction `cv.imwrite("nom.bmp",img)`. Quel est le poids du fichier obtenu sur l'espace de stockage ? Comparer avec son poids théorique.

Manipulation 3

1. Lire et visualiser l'image « mandrill.png » à l'aide des fonctions `cv.imread` et `plt.imshow`

```
img = cv.imread('/path_to_image/image.jpg')
```

```
plt.imshow(img)
```

Remarque : Dans le cas des images en couleur, la fonction `cv.imread` stocke les canaux RGB dans l'ordre inverse BGR, il faut donc les intervertir avec l'instruction `cv.cvtColor(img, cv.COLOR_BGR2RGB)`.

2. Afficher la taille et la profondeur de l'image avec les fonctions `img.shape` et `img.dtype`.
3. Pourquoi les valeurs de l'image sont de type « uint8 » ?
4. Séparer les canaux RGB et visualiser chaque canal indépendamment à l'aide de l'instruction `cv.split`.
5. Expliquer comment apparaissent les yeux du mandrill sur les 3 composantes RGB.
6. Intervertir l'ordre des canaux RGB en faisant toutes les combinaisons possibles. Visualiser et commenter
7. Stocker la composante R de l'image dans un fichier .png, la composante G dans un fichier .bmp et la composante B dans un fichier .jpeg en utilisant l'instruction `cv.imwrite`
8. Calculer le poids théorique de chaque composante (R, G, B) et comparer avec le poids des fichiers obtenus à la question précédente.
9. Réaliser un passage en niveau de gris en utilisant l'expression suivant :

$$luminance = 0,2126 \times Rouge + 0,7152 \times Vert + 0,0722 \times Bleu.$$

10. Transformer l'image du mandrill en niveaux de gris en utilisant la fonction `cv.cvtColor(img, cv.COLOR_BGR2GRAY)`. Obtient-on le même résultat que précédemment ? Commenter.