

<b>TD INFO08</b> <b>MANIPULATION D'IMAGES</b>
--

Les images sont stockées dans des fichiers binaires. Plusieurs formats d'images existent jpeg, gif, png, bmp...

La librairie `matplotlib` permet de traiter facilement les images au format png.

On utilise le fichier `ara.png`, le placer dans votre répertoire de travail.

```
# Ouverture d'un fichier png
import matplotlib.pyplot as plt
image = plt.imread("ara.png")

# Affichage
plt.imshow(image)
plt.show()
```

La bibliothèque `matplotlib` décode le fichier et le stocke sous forme d'un tableau `numpy`.  
Le tableau obtenu possède trois dimensions : ligne, colonne, couleur.

```
# Récupération des dimensions
(hauteur, largeur, couleur) = image.shape
```

Chaque couleur est codée selon le modèle RGB (Red-Green-Blue) : 3 réels compris entre 0 et 1 codent respectivement les quantités de rouge, vert et bleu.

```
image[i, j, 0] correspond à la valeur de rouge du pixel situé à la ligne i et la colonne j
image[i, j, 1] correspond à la valeur de vert du pixel situé à la ligne i et la colonne j
image[i, j, 2] correspond à la valeur de bleu du pixel situé à la ligne i et la colonne j
```

<b>Exercice n°1 : Filtres rouge, vert et bleu</b>
---

On va afficher les images obtenues par application d'un filtre rouge (qui ne laisse passer que le rouge) ou d'un filtre vert ou d'un filtre bleu.

Pour cela, on ouvre le fichier image et on récupère les dimensions de l'image.

Puis on initialise un tableau `numpy` à 3 dimensions (hauteur, largeur, 3) vide (c'est à dire ne contenant que des 0) à l'aide de la commande `np.zeros(( , , ))`.

Ensuite on copie les valeurs utiles (il n'est pas utile d'utiliser des boucles `for`, on fait de l'extraction de valeurs de tableaux).

Enfin on affiche l'image.

### Exercice n°2 : Niveaux de gris

Pour obtenir une image en niveaux de gris les trois valeurs RGB doivent être identiques ([0, 0, 0] : noir ; [1, 1, 1] : blanc).

Il existe plusieurs algorithmes pour transformer une image couleur en niveaux de gris :

« gris moyen » : le niveau de gris est la moyenne des niveaux de rouge, vert et bleu ;

« gris clarté » : le niveau de gris est la moyenne entre le maximum et le minimum des niveaux de rouge, vert et bleu ;

« gris luminosité » : le niveau de gris est obtenu par la formule  $0,21 \times R + 0,71 \times G + 0,07 \times B$ .

Afficher et comparer les images obtenues par chacun des traitements.

### Manipulation d'images en niveaux de gris

Il y a plusieurs façons de procéder :

#### # avec deux boucles for pour parcourir chaque pixel

```
G_moyen = np.zeros((hauteur, largeur, 3))
for i in range(hauteur):
    for j in range(largeur):
        gris = (image[i, j, 0] + image[i, j, 1] + image[i, j, 2]) / 3
        G_moyen[i, j] = [gris, gris, gris]
plt.figure("gris moyen")
plt.imshow(G_moyen)
plt.show()
```

#### # en manipulant directement les tableaux numpy relatifs à chaque couleurs

```
R = image[:, :, 0]
G = image[:, :, 1]
B = image[:, :, 2]

G_moyen = np.zeros((hauteur, largeur, 3))
Gris = (R+G+B) / 3
G_moyen[:, :, 0] = Gris
G_moyen[:, :, 1] = Gris
G_moyen[:, :, 2] = Gris
plt.figure("gris moyen")
plt.imshow(G_moyen)
plt.show()
```

```
# en utilisant cmap="gray"

R = image[:, :, 0]
G = image[:, :, 1]
B = image[:, :, 2]

Gris = (R+G+B)/3
plt.figure("gris_moyen")
plt.imshow(Gris, cmap="gray")
plt.show()
```

Remarquer que le tableau `Gris` ne contient qu'une seule valeur (le niveau de gris). Pour l'affichage, il est obligatoire de spécifier `cmap="gray"` afin que Python gère correctement l'affichage. Il est plus commode de manipuler un tableau à 2 dimensions plutôt que trois tableaux à 2 dimensions identiques.

### Exercice n°3 : Noir et blanc

À partir de l'image en niveaux de gris « luminosité », proposer un programme qui crée une image en noir et blanc.

### Exercice n°4 : Miroir

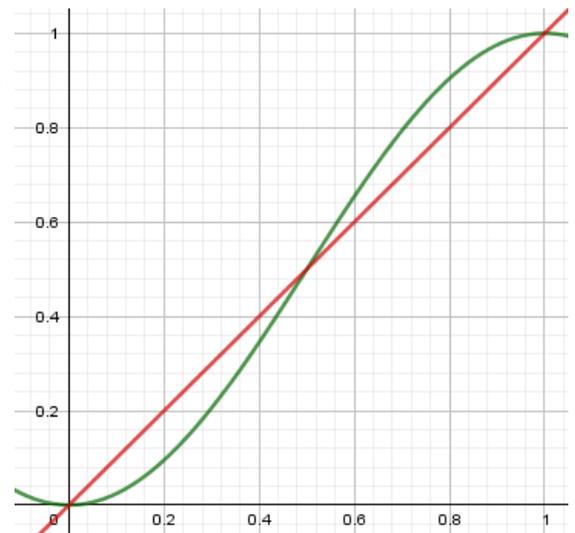
À partir d'une image, créer son image miroir (verticalement).

### Exercice n°5 : Amélioration du contraste

À partir de l'image en niveaux de gris, créer une nouvelle image en niveaux de gris en appliquant à chaque valeur de niveaux de gris la fonction :

$$f(x) = 0,5 + 0,5 \times \sin(\pi(x - 0,5))$$

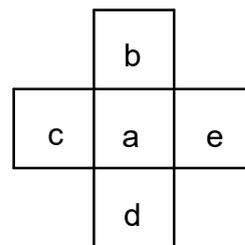
Expliquer pourquoi cela améliore le contraste.



### Exercice n°6 : Détection des bords

Le bord d'un objet correspond à une zone où la valeur du niveau de gris varie rapidement.

Pour savoir si le pixel avec une valeur  $a$  est le long d'un bord d'un objet, on prend en compte les valeurs  $b$ ,  $c$ ,  $d$  et  $e$  de ses quatre voisins horizontaux et verticaux.



On calcule la valeur  $t$  suivant la formule :  $t = \sqrt{(b-d)^2 + (c-e)^2}$ .

Si le pixel n'est pas proche d'un bord,  $b \approx d$  et  $c \approx e$  alors  $t$  est « faible » (proche de 0).

En revanche si le pixel est proche d'un bord alors  $t$  est « grand ».

À partir d'une image en niveaux de gris, créer une image en noir en blanc en affichant du blanc quand  $t$  est inférieur à un certain seuil et du noir quand il est supérieur. La valeur seuil de  $t$  sera à optimiser.