# Traitement d'image : niveaux de gris

## 1. Introduction

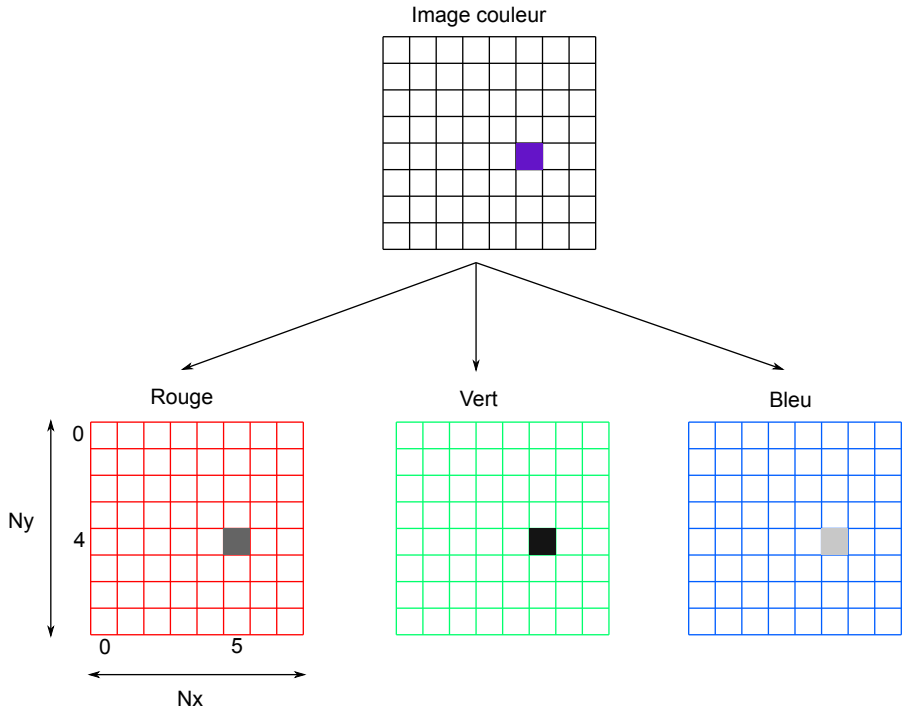
Ce document est une introduction au traitement des images avec Python. Il explique comment lire une image dans un fichier, extraire les couches d'une image couleur, et faire des manipulations élémentaires sur les niveaux de gris. On verra enfin comment enregistrer une image dans un fichier.

## 2. Lecture d'une image

### 2.a. Codage d'une image en couleur

Les images fournies par les appareils photo sont généralement en couleur. Une image en couleur est contituée de trois couches : une couche rouge (R), une couche verte (V), une couche bleue (B). Nous n'allons pas ici expliquer comment sont représentées les couleurs. Voir à ce sujet : Espace des couleurs RVB et la simulation Espace des couleurs RVB et triangle de Maxwell.

Soit $N_x$ le nombre de colonnes de l'image et $N_y$ le nombre de lignes. Le nombre de pixels total est $N = N_x N_y$. Chaque couche est une matrice comportant $N_y$ lignes et $N_x$ colonnes. Le plus souvent, cette matrice contient des entiers codés sur 8 bits (les valeurs vont de 0 à 255). Pour l'image en couleur complète, il y a donc 24 bits par pixels, à multiplier par le nombre de pixels pour obtenir l'occupation totale en mémoire. Chaque couche peut être vue comme une image en niveaux de gris. Le niveau 0 est le noir, le niveau 255 est le blanc, le niveau 128 est un gris moyen.



On voit sur la figure le pixel $(5, 4)$ dont les niveaux de gris des trois couches sont $(100, 20, 200)$, ce qui donne une couleur violette. Par convention, on notera en premier l'indice qui repère les colonnes, conformément au repérage des coordonnées d'un point sur un plan $(x, y)$. On remarque néanmoins que l'origine se trouve sur le coin supérieur gauche de l'image.

### 2.b. Lecture d'un fichier image

Voici l'image en couleur :

Pour lire un fichier d'image (PNG, JPEG, etc), on utilise le module imageio :

```
import imageio
import numpy
from matplotlib.pyplot import *
img = imread("../../../../figures/image/niveaux/images/babouin.png")
```

Voici les dimensions du tableau et le type de données :

```
print(img.shape)
--> (256, 256, 3)

print(img.dtype)
--> dtype('float32')
```

La valeur d'un pixel pour une couche est un flottant compris entre 0 et 1, obtenu en divisant la valeur entière (8 bits) par 256.

La séparation des trois couches se fait par :

```
rouge = img[:,:,0]
vert = img[:,:,1]
bleu = img[:,:,2]
```

On se contentera ici de travailler sur la couche rouge, qui est une image en niveaux de gris. On peut voir la valeur d'un pixel particulier, celui d'indices $(120, 100)$ :

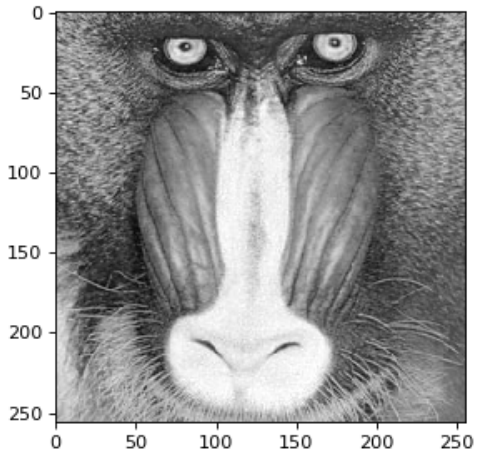
```
print(rouge[100,120])
--> 0.93333334
```

Remarquons que le premier indice d'accès au tableau indique la ligne.

### 2.c. Affichage d'une image

La fonction `matplotlib.pyplot.imshow` permet d'afficher un tableau sous forme d'une image :

```
figure(figsize=(4,4))
imshow(rouge,cmap='gray')
```

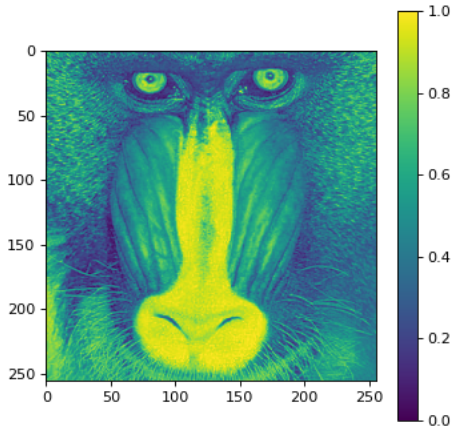



Le pixel de coordonnées $(i, j)$ est l'élément `rouge[j,i]`.

L'argument `cmap='gray'` permet de représenter les valeurs en niveaux de gris. La fonction `matplotlib.pyplot.imshow` ainsi utilisée convertit la plage de valeurs $[min, max]$ du tableau en valeurs dans l'intervalle $[0, 255]$ pour l'affichage en image.

Il peut être intéressant de représenter l'image en niveaux de gris avec une échelle de couleurs, car nous identifions plus facilement une couleur qu'un niveau de gris :

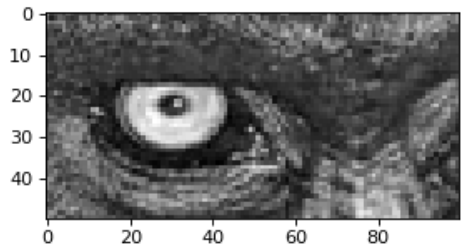
```
figure(figsize=(5,5))
imshow(rouge)
colorbar()
```




### 2.d. Extraction d'une partie d'une image

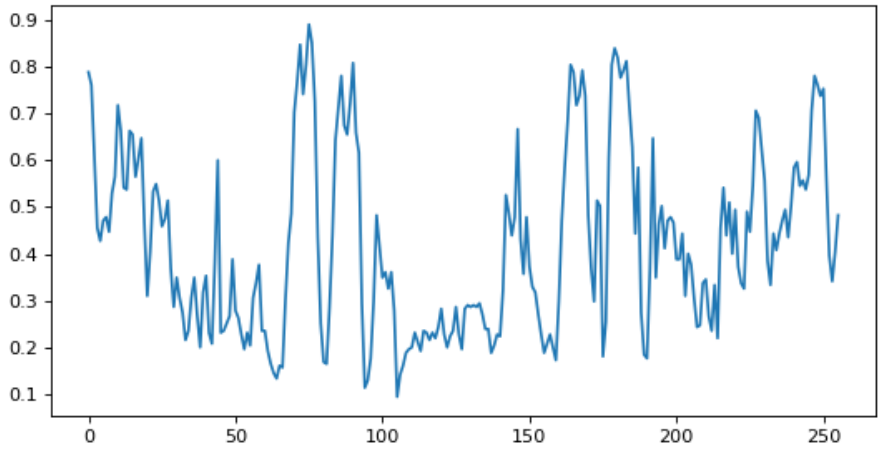
Une partie rectangulaire d'une image peut être extraite de la manière suivante, en faisant attention à indiquer les lignes sélectionnées en premier :

```
partie = rouge[0:50,50:150]
figure(figsize=(4,4))
imshow(partie,cmap='gray')
```




Une opération courante est la sélection d'une seule ligne (ou d'une seule colonne) :

```
ligne20 = rouge[20,:]
figure(figsize=(8,4))
plot(ligne20)
```




On obtient ainsi le profil de niveaux de gris sur cette ligne.

## 3. Étude statistique des niveaux de gris

Voici comment obtenir la moyenne des valeurs d'une image et l'écart-type :

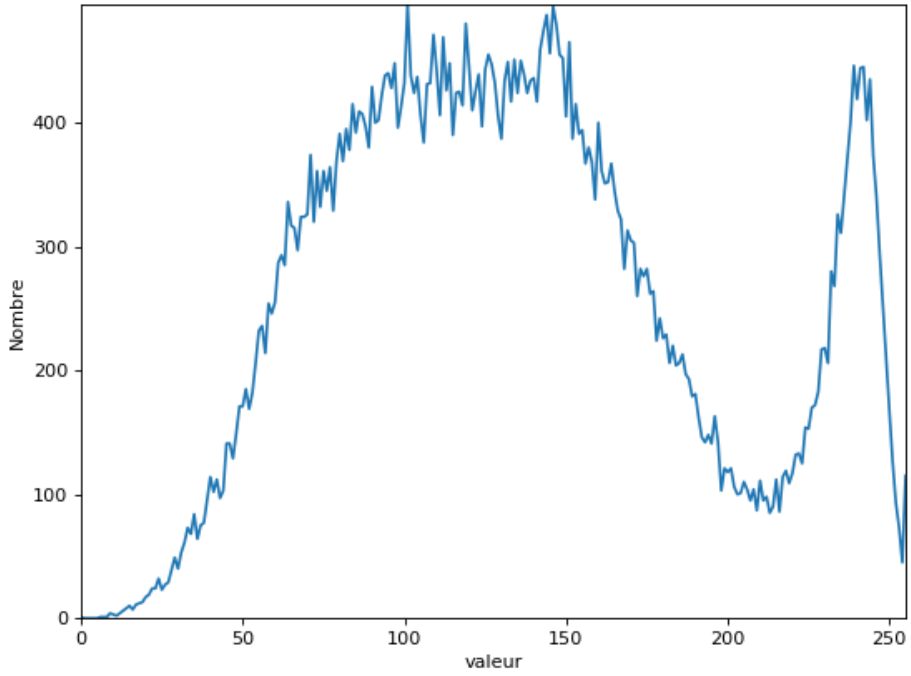
```
print(rouge.mean())
--> 0.53763819

print(rouge.std())
--> 0.22340311
```

Pour obtenir plus d'informations, on peut calculer un histogramme. Pour chaque valeur entière comprise entre 0 et 255, on compte le nombre de pixels qui ont cette valeur.

```
def histogramme(image):
    h = numpy.zeros(256,dtype=numpy.float32)
    s = image.shape
    for j in range(s[0]):
        for i in range(s[1]):
            valeur = image[j,i]
            h[int(valeur*255)] += 1
    return h

h = histogramme(rouge)
figure(figsize=(8,6))
plot(h)
axis([0,255,0,h.max()])
xlabel("valeur")
ylabel("Nombre")
```

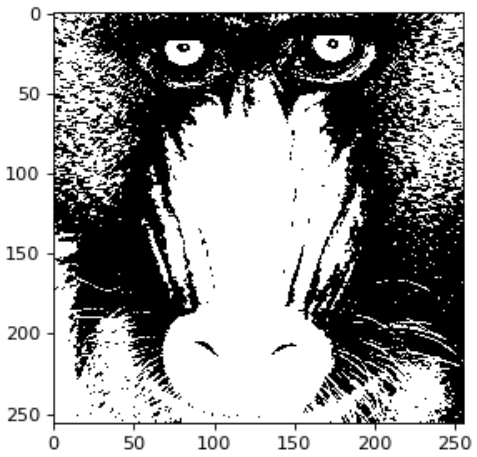



## 4. Modification des niveaux de gris

Une opération courante de traitement d'image est le seuillage, qui consiste à repérer les pixels dont la valeur dépasse un certain seuil.

```
def seuillage(image,seuil):
    resultat = image.copy()
    s = image.shape
    for j in range(s[0]):
        for i in range(s[1]):
            if image[j,i] > seuil:
               resultat[j,i] = 1
            else:
                resultat[j,i] = 0
    return resultat

im4 = seuillage(rouge,0.5)
figure(figsize=(4,4))
imshow(im4,cmap='gray',vmin=0,vmax=1)
```

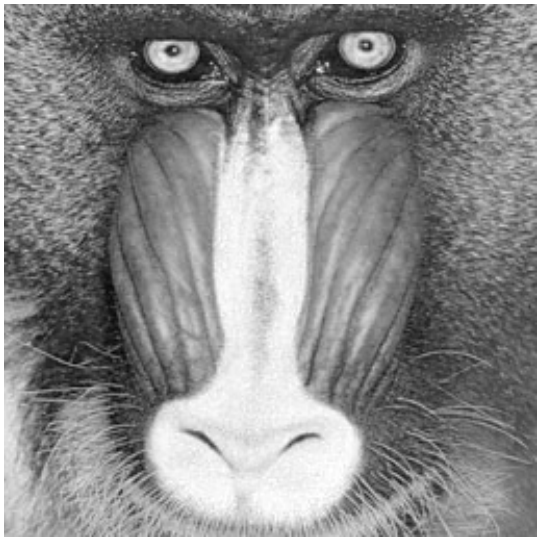



## 5. Enregistrement d'une image

L'image précédente peut être enregistrée avec la fonction `imageio.imwrite` :

```
imageio.imwrite("../../../../figures/image/niveaux/images/imA.png",rouge)
```

Voici l'image obtenue avec ce fichier :



Il s'agit d'une image en couleur dont les trois couches sont identiques, ce qui donne des niveaux de gris.

On peut aussi reconstituer une image en couleur à partir de ses trois couches. Pour obtenir une image différente de l'image initiale, on réduit les valeurs de la couche rouge. On doit tout d'abord créer un tableau à trois dimensions et le remplir avec les couches.

```
s = rouge.shape
couleur = numpy.zeros((s[0],s[1],3),dtype=numpy.float32)
couleur[:,:,0] = rouge*0.5
```

```
couleur[:,:,1] = vert
couleur[:,:,2] = bleu
imsave("../../../../figures/image/niveaux/images/imB.png",couleur)
```