

# Traitement d'image avec commentaires

(compléments d'informations à destination du professeur)

Raoul HATTERER

7 février 2019

## Table des matières

<b>1</b>	<b>Installation de PIL</b>	<b>2</b>
<b>2</b>	<b>Codage RVB et niveau de gris</b>	<b>2</b>
<b>3</b>	<b>Image de départ</b>	<b>2</b>
<b>4</b>	<b>Comment lire un pixel</b>	<b>2</b>
<b>5</b>	<b>Comment écrire un pixel</b>	<b>3</b>
<b>6</b>	<b>Que fait le programme suivant ?</b>	<b>3</b>
<b>7</b>	<b>Passage d'une image en niveau de gris</b>	<b>3</b>
<b>8</b>	<b>Passage d'une image en vrai niveau de gris (sans informations triplées)</b>	<b>5</b>
8.1	Utilisation du mode L (luminance) pour les images en nuances de gris= . . . . .	5
8.2	Existe-t-il d'autres modes ? . . . . .	5
<b>9</b>	<b>Pour aller plus loin</b>	<b>6</b>
9.1	Créer une image en négatif . . . . .	6
9.2	Diagonale . . . . .	7

Source : [Traitement d'image de l'académie de grenoble](#)

## 1 Installation de PIL

À faire au préalable par le professeur.

```
pip3 install pillow
```

## 2 Codage RVB et niveau de gris

Aller sur [colors RGB](#) et tester ce que l'on obtient si l'on remplace chacune des valeurs R, V et B d'un pixel par la moyenne des sous-pixels. Essayer pour plusieurs couleurs.

## 3 Image de départ



FIGURE 1 – Image de départ (480px × 300px)

## 4 Comment lire un pixel

Après avoir fait quelques recherches sur ce qu'est un "pixel", voyons comment lire le pixel de coordonnées (100,250).

```

from PIL import Image
img = Image.open("pomme.jpg")
r,v,b=img.getpixel((100,250))
print("canal rouge : ",r,"canal vert : ",v,"canal bleu : ",b)

canal rouge :  19 canal vert :  88 canal bleu :  192

```

## 5 Comment écrire un pixel

```

from PIL import Image
img = Image.open("pomme.jpg")
img.putpixel((5,5),(255,0,0))
img.show()

```

## 6 Que fait le programme suivant ?

```

from PIL import Image                                     # Importation de la librairie PILLOW (gestion d'images)
img = Image.open("pomme.jpg")                             # Mise en mémoire dans la variable "img" de l'image "pomme.jpg"
largeur_image,hauteur_image=img.size                     # Méthode size appliquée à la variable img pour obtenir sa largeur et sa hauteur

for y in range(hauteur_image):                             # Boucle pour parcourir les toutes les lignes de l'image
    for x in range(largeur_image):                         # Boucle imbriquée pour parcourir les pixels de l'image
        rouge,vert,bleu=img.getpixel((x,y))               # Méthode getpixel appliquée à la variable img pour obtenir les valeurs des canaux rouge, vert et bleu
        nouveau_rouge=vert                                # Le vert devient rouge
        nouveau_vert=bleu                                  # Le bleu devient vert
        nouveau_bleu=rouge                                # Le rouge devient bleu
        img.putpixel((x,y),(nouveau_rouge,nouveau_vert,nouveau_bleu)) # Méthode putpixel appliquée à la variable img pour modifier les valeurs des canaux rouge, vert et bleu

img.show()                                                 # Affichage de l'image obtenue
img.save("pommeMystere.jpg")                             # Sauvegarde de l'image obtenue

```

On analyse le code ci-dessus qui servira de base pour le défi suivant.

## 7 Passage d'une image en niveau de gris

Après avoir fait quelques recherches sur les "images en niveau de gris", écrivez un programme qui transforme une "image couleur" en une "image en niveau de gris".



FIGURE 2 – Résultat du programme mystère

Petite astuce qui pourrait vous aider : en Python pour avoir une division entière (le résultat est un entier), il faut utiliser l'opérateur `//` à la place de l'opérateur `/`

Remarque : On donne l'algorithme aux élèves (ou on le construit avec eux); ils doivent alors programmer le passage d'une image couleur à une image en niveaux de gris.

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image=480
hauteur_image=300

for y in range(hauteur_image):
    for x in range(largeur_image):
        rouge,vert,bleu=img.getpixel((x,y))
        nouveau_rouge=(vert+bleu+rouge)//3
        nouveau_vert=(vert+bleu+rouge)//3
        nouveau_bleu=(vert+bleu+rouge)//3
        img.putpixel((x,y),(nouveau_rouge,nouveau_vert,nouveau_bleu))

img.show()
img.save("pommegrise.jpg")
```

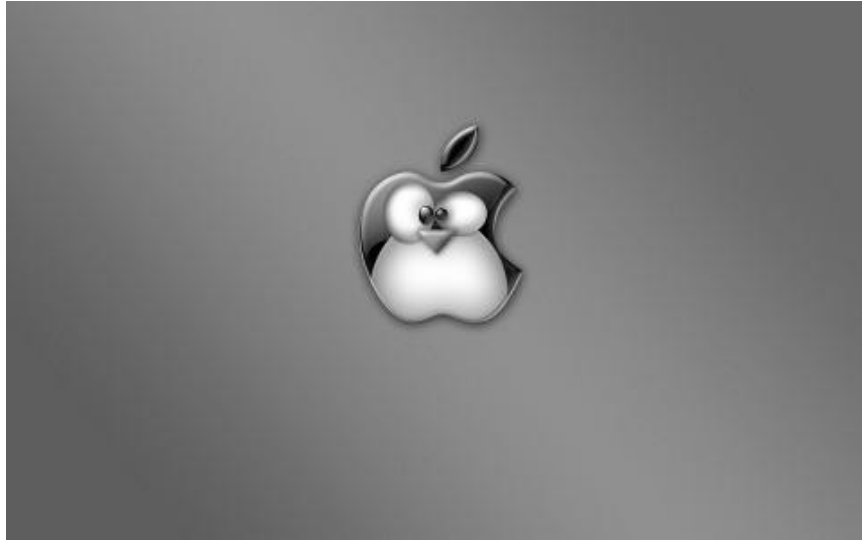


FIGURE 3 – Image en niveaux de gris

## 8 Passage d'une image en vrai niveau de gris (sans informations triplées)

### 8.1 Utilisation du mode L (luminance) pour les images en nuances de gris=

```
from PIL import Image
img = Image.open("pomme.jpg").convert("L")
img.show()
img.save("pommegriseL.jpg")
```

Comparer la taille des différents fichiers. Conclure.

*Réponse : code sur un octet par pixel l'image prend moins de place donc le fichier est moins lourd. Remarque : Pas trois fois moins lourd ; notamment car le format jpeg est compressé.*

### 8.2 Existe-t-il d'autres modes ?

Les **modes** supportés par Pillow sont :

- 1 (1-bit pixels, black and white, stored with one pixel per byte)
- L (8-bit pixels, black and white)



FIGURE 4 – Image en niveaux de gris (sans redondance)

- P (8-bit pixels, mapped to any other mode using a color palette)
- RGB (3x8-bit pixels, true color)
- RGBA (4x8-bit pixels, true color with transparency mask)
- CMYK (4x8-bit pixels, color separation)
- YCbCr (3x8-bit pixels, color video format)
- LAB (3x8-bit pixels, the L\*a\*b color space)
- HSV (3x8-bit pixels, Hue, Saturation, Value color space)
- I (32-bit signed integer pixels)
- F (32-bit floating point pixels)

## 9 Pour aller plus loin

### 9.1 Créer une image en négatif

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image, hauteur_image = img.size

for y in range(hauteur_image):
    for x in range(largeur_image):
        rouge, vert, bleu = img.getpixel((x, y))
```

```

nouveau_rouge=255-rouge
nouveau_vert=255-vert
nouveau_bleu=255-bleu
img.putpixel((x,y),(nouveau_rouge,nouveau_vert,nouveau_bleu))

img.show()
img.save("pommeNegatif.jpg")

```



FIGURE 5 – Négatif

## 9.2 Diagonale

Créer le programme qui garde l'image d'origine au-dessus d'une diagonale et qui transforme en niveau de gris en-dessous

```

from PIL import Image
img = Image.open("pomme.jpg")
largeur_image,hauteur_image=img.size

for y in range(hauteur_image):
    tailleDiag=int((largeur_image/hauteur_image)*y)
    for x in range(tailleDiag):
        r,v,b=img.getpixel((x,y))

```

```
n_r=(v+b+r)//3
n_v=(v+b+r)//3
n_b=(v+b+r)//3
img.putpixel((x,y),(n_r,n_v,n_b))

img.show()
img.save("pommemisgrise.jpg")
```

