

Traitement d'image

(1 séance = 1h30)

Raoul HATTERER

7 février 2019

Table des matières

1	Installation de PIL	1
2	Codage RVB et niveau de gris	2
3	Image de départ	2
4	Comment lire un pixel	2
5	Comment écrire un pixel	3
6	Que fait le programme suivant ?	3
7	Passage d'une image en niveau de gris	3
8	Passage d'une image en vrai niveau de gris (sans informations triplées)	4

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
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 = vert
        nouveau_vert = bleu
        nouveau_bleu = rouge
        img.putpixel((x,y), (nouveau_rouge, nouveau_vert, nouveau_bleu))
img.show()
img.save("pommeMystere.jpg")
```

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".

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
```



FIGURE 2 – Résultat du programme mystère

```
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("pommeprise.jpg")
```

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

```
from PIL import Image
img = Image.open("pomme.jpg")
largeur_image,hauteur_image=img.size
img2 = Image.new("L",(largeur_image,hauteur_image))

for y in range(hauteur_image):
    for x in range(largeur_image):
```

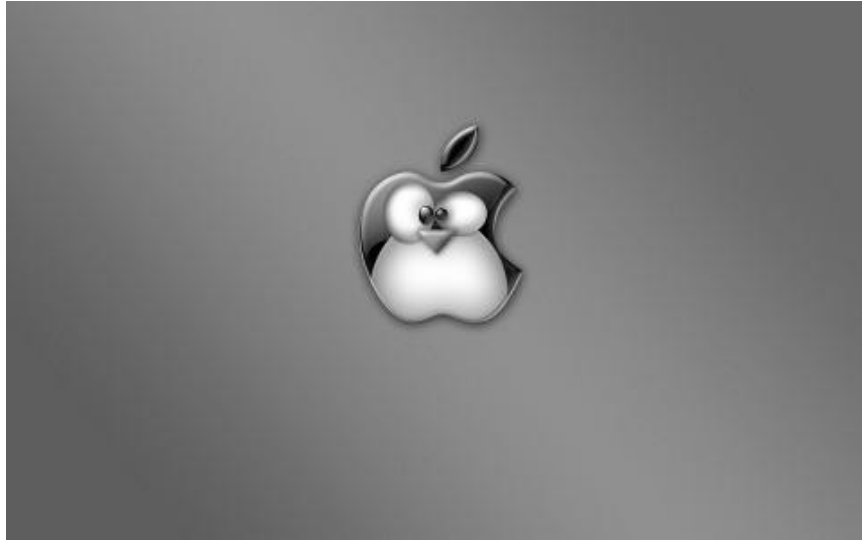


FIGURE 3 – Image en niveaux de gris

```
rouge,vert,bleu=img.getpixel((x,y))
niveau_gris=(vert+bleu+rouge)//3
img2.putpixel((x,y),(niveau_gris))

img2.show()
img2.save("pommegriseL.jpg")
```

Comparer la taille des différents fichiers. Conclure.



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