Chapitre # (ALGO) 1 Traitement numérique d'images en niveaux de gris

1 Introduction et commandes principales .....

2 Traitement d'image et estimation d'une proportion .....

3 Solutions des exercices .....

Résumé & Plan

Ce court chapitre vise à utiliser des outils de traitement d'images afin de déterminer une proportion de bactéries dans une boîte de Pétri.

- Les chapitres d'Informatique sont composés de cours et d'exercices intégrés. Le cours sera projeté au tableau.
- Il n'est pas attendu que toute la classe aborde tous les exercices. Traitez donc en priorité les exercices présents dans la liste donnée à chaque début de séance.
- Exercices ◐ / **Pour aller plus loin** : exercices plus difficiles, ou plus techniques. À ne regarder que si les autres sont bien compris.

1. INTRODUCTION ET COMMANDES PRINCIPALES

Les appareils numériques fournissent des photos en couleurs (codage RVB) ou en noir et blanc (codage en niveaux de gris). Nous étudierons ici uniquement les images en noir et blanc.

Une image numérique en « noir et blanc » est un tableau de valeurs. Chaque case de ce tableau est un pixel qui stocke une valeur de gris . En notant n le nombre de lignes et p le nombre de colonnes de l'image, on manipule ainsi un tableau de $n \times p$ pixels. Chaque pixel est repéré par ses coordonnées (x, y) : le point de coordonnées $(0, 0)$ est en haut à gauche, x représente l'indice de colonne et y l'indice de ligne.
Les valeurs des pixels sont enregistrées sous forme de nombres entiers entre 0 et 255 qui représentent la luminosité, ce qui fait 256 valeurs possibles pour chaque pixel.

La valeur 0 correspond au noir (aucune lumière), et la valeur 255 correspond au blanc (luminosité maximale). Les valeurs intermédiaires correspondent à des niveaux de gris allant du plus foncé au plus clair.

Module et fonctions Python utiles pour la manipulation des images :

On importe le module consacré aux images par : `import PIL.Image as Image`

- Pour ouvrir un fichier image existant : `img=Image.open("monfichier.jpg")`

Attention

Le fichier doit être enregistré dans le répertoire courant (c'est-à-dire dans le même dossier que le script)

On peut alors effectuer différentes actions sur l'objet `img` :

`img.show()` → affiche l'image dans le programme par défaut.

`img.size` → couple (largeur,hauteur) de l'image (en nombre de pixels).

`img.getpixel((x,y))` → donne la « couleur » (niveau de gris) du pixel (x, y), c'est un entier compris entre 0 et 255.

`img.putpixel((x,y),couleur)` → met le pixel (x, y) à la « couleur » indiquée qui est un entier compris entre 0 et 255.

- Pour créer une nouvelle image :

```
nouvelle=Image.new(mode,(largeur,hauteur),color=couleur)
```

Si `mode = "L"` , l'image est codée en niveaux de gris de 0 à 255.

Si `mode= "1"`, l'image est codée uniquement en noir et blanc (c'est-à-dire 0 ou 255). « couleur » est la couleur de fond, c'est-à-dire une valeur numérique entre 0 et 255. La nouvelle image est stockée dans la variable `nouvelle`

- Pour sauvegarder cette image : `nouvelle.save("nomfichier.png",format="png")`
- Pour convertir une image en couleurs en niveaux de gris : `img.convert('L')`.

2. TRAITEMENT D'IMAGE ET ESTIMATION D'UNE PROPORTION

Pour ce TP, on récupèrera la photo « petri1.png » sur ma page web (sélectionnez la photo, clic droit pour enregistrer sous. Enregistrez la photo sous le nom « petri.png », dans le même dossier que votre fichier Python).

Exercice 1 | Manipulation de « petri1 » Solution

Il s'agit de la photo en noir et blanc (c'est-à-dire en niveaux de gris) d'une boîte de pétri faisant apparaître un développement de bactéries. Dans cet exercice, après quelques manipulations de l'image, on cherchera à déterminer une estimation de la surface colonisée par les bactéries.

1. Créer un script Python dans lequel :
→ on importe le module `PIL.Image`
→ on ouvre le fichier `"petri1.jpg"` sous le nom `img`
→ on détermine la hauteur et la largeur de la photo sous les noms`hauteur` et `largeur`.
Enregistrer ce fichier Python et la photo petri1 dans un MÊME répertoire

2. Tester alors, à partir du script précédemment créé, les instructions suivantes :

```
>>> img.show()
>>> hauteur
>>> largeur
>>> img.getpixel((250,270))
>>> img.getpixel((20,10))
>>> img.putpixel((20,10),0)
>>> img.show()
```

3. [Visualisation de la colonie de bactéries]

3.1) Ecrire une fonction `moyenne()` qui retourne la valeur moyenne des couleurs des pixels de l'image « petri1 ». Tester :

```
>>> moyenne()
```

3.2) Ecrire une fonction `seuillage()` qui crée à partir de l'image « petri1.jpg » une nouvelle image « petri1nb.png » formée uniquement de pixels noirs et blancs. Le pixel de la nouvelle image sera noir lorsque la valeur du pixel original est inférieure à la moyenne (gris foncé à noir) et blanc lorsque la valeur du pixel original est supérieure (blanc à gris clair). Tester :

```
>>> seuillage()
```

La nouvelle image « petri1nb.png » est automatiquement enregistrée dans le répertoire courant.
On obtient ainsi une version très contrastée de la photo originale.
Les bactéries apparaissent alors en blanc sur le fond noir de la boîte, le fond de la photo est blanc.
On constate que cette image fait apparaître plus de blanc à l'intérieur de la boîte que la surface réellement colonisée par les bactéries. Comment peut-on expliquer ce phénomène ?

3.3) Modifier cette fonction sous la forme `seuillage(m)` afin de « seuiller » l'image à partir d'une valeur m choisie par l'utilisateur.

3.4) Quelle valeur choisissez-vous afin d'obtenir une représentation relative-mensatisfaisante de la colonie de bactéries ?

4. [Estimation de la surface colonisée par les bactéries]
Nous allons maintenant déterminer la surface totale de la boîte puis la surface colonisée par les bactéries.
Pour déterminer la surface totale de la boîte, il faut déterminer son rayon, ou plutÙt son diamètre. *(L'unité utilisée sera le pixel.)*

4.1) En comptant le nombre de colonnes blanches sur la photo, écrire une fonction `diametre()` qui retourne le diamètre de la boîte. On pourra introduire une fonction `colonne_bl(i)` qui renvoie `True` si la i-ème colonne est blanche.

```
>>> diametre()
```

4.2) Ecrire ensuite une fonction `surface()` qui retourne une estimation de la surface totale de la boîte.

4.3) Ecrire une fonction `surface_noire()` qui retourne le nombre total de pixels noirs de l'image « petri1nb.png », c'est-à-dire la surface non colonisée par les bactéries.

4.4) Ecrire enfin une fonction `proportion()` qui retourne le pourcentage de la surface de la boîte colonisée par les bactéries.

```
>>> proportion()
```

Exercice 2 | Le blanc devient le noir, et vise-versa Solution Ecrire une fonction `negatif()` qui crée une image « petrineg.jpg » obtenue en échangeant les valeurs sombres et claires de l'image « petri1.jpg ».

3. SOLUTIONS DES EXERCICES

Solution (exercice 1) Énoncé

```
## Question 1

import PIL.Image as Image

img = Image.open("petri1.jpg")

largeur, hauteur = img.size

## Question 3.(1)

def moyenne():
    S = 0
    for i in range(largeur):
        for j in range(hauteur):
            S = S + img.getpixel((i,j))
    return S/(hauteur*largeur)

## Question 3.(2)

def seuillage():
    nouvelle = Image.new("1",(largeur,hauteur),color=0) # \
    ↪ Créé une image noire
    for i in range(largeur):
        for j in range(hauteur):
            pixel = img.getpixel((i,j))
            if pixel < 157: #La moyenne trouvée précédemment
                pixel = 0  # Le point est ramené en noir
            else :
                pixel = 255
            nouvelle.putpixel((i,j),pixel)
    nouvelle.save('petri1nb.png',format='png')
```

```
## Question 3.(3)

def seuillage(m):
    nouvelle=Image.new("1",(largeur,hauteur),color=0) # Créé \
    ↪ une image noire
    for i in range(largeur):
        for j in range(hauteur):
            pixel = img.getpixel((i,j))
            if pixel < m:
                pixel = 0  # le point est ramené en noir
            else :
                pixel = 255
            nouvelle.putpixel((i,j),pixel)
    nouvelle.save('petri1nb2.png',format='png')

# On propose de prendre m = 180

## Question 4.(1)

def colonne_bl(i): #renvoie True si la iième colonne est \
↪ blanche
    image=Image.open('petri1nb2.png')
    j=0
    while j<hauteur and image.getpixel((i,j)) == 255:
        j=j+1
    return j==hauteur

def diametre():
    compteur=0
    for i in range(largeur):
        compteur=compteur+colonne_bl(i)
    return(largeur-compteur)

def rayon(): # determine le rayon
    return diametre()/2
```

```python
## Question 4.(2)

from math import pi

def surface():
    return (rayon()**2)*pi

## Question 4.(3)

def surface_noire(): # determine le nombre de pixels noirs
    image=Image.open('petri1nb2.png')
    s=0
    for i in range(largeur):
        for j in range(hauteur):
            if image.getpixel((i,j)) == 0:
                s=s+1
    return s

## Question 4.(4)

def proportion(): # determine la proportion de surface de la \
↪ boîte occupee par les bacteries (le blanc)
    total=surface()
    return ((total-surface_noire())/total)
```

Solution (exercice 2) Énoncé

```python
def negatif(): # crée le negatif de l'image
    img2=Image.new("L",(largeur,hauteur),color=0)
    for i in range(largeur):
        for j in range(hauteur):
            img2.putpixel((i,j),255-img.getpixel((i,j)))
    img2.save('petrineg.png',format='png')
```