Département MI Centre Universitaire de Mila

Traitement d'images Dr. B Maamar

Travaux pratiques N°1

1. Objectifs

• Introduire le langage de programmation Python.

• Introduire les commandes de base de traitement d'image avec Python et se familiariser avec quelques bibliothèques de traitement d'image comme :

• **NumPy** prend en charge les tableaux

• **PIL/Pillow (Python Imaging Library)** est une bibliothèque open-source Python, qui prend en charge l'ouverture, la manipulation et l'enregistrement de nombreux formats de fichiers d'image différents. Image

• **Image Scikit** utilise des tableaux NumPy comme objets image, propose de nombreux algorithmes différents pour la segmentation, la manipulation de l'espace colorimétrique, la transformation géométrique, l'analyse, la morphologie, la détection de caractéristiques, et bien plus encore.

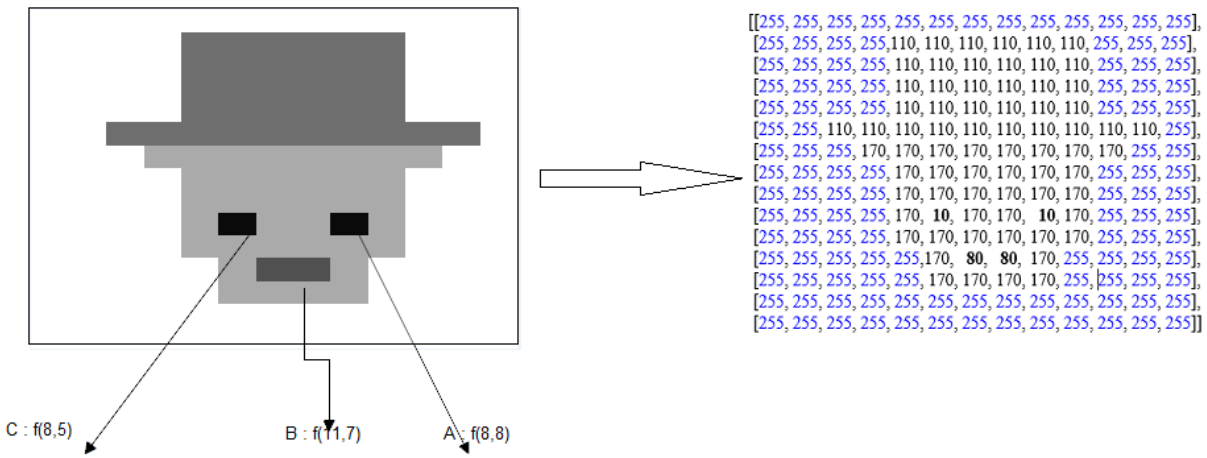
• **Scipy** est un autre des modules scientifiques de base de Python comme Numpy et peut être utilisé pour des tâchOpenCV-Python

• **OpenCV (Open Source Computer Vision Library)** est l'une des bibliothèques les plus utilisées pour les applications de Computer Visiones de base de manipulation et de traitement d'images.

• **Matplotlib** est une autre excellente option pour une bibliothèque de traitement d'image Matplotlib est spécialisé dans les tracés 2D de tableaux en tant que bibliothèque de visualisation de données multiplateforme sur des tableaux Numpy

2. Enoncé

Soit l'image suivante et la matrice qui la représente en niveaux de gris.



Créer le tableau 2D ci-dessus.

Rép :

```
pip install numpy

import numpy as np

Im_01 =  np.array( [[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 110, 110, 110, 110, 110, 110, 110, 110, 110, 110, 255],
                    [255, 255, 255, 170, 170, 170, 170, 170, 170, 170, 170, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255, 255, 255, 255, 170,  10,  170, 170,  10,  170, 255, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255, 255, 255, 255, 255, 170,  80,  80,  170, 255, 255, 255, 255],
                    [255, 255, 255, 255, 255, 170, 170, 170, 170, 255, 255, 255, 255],
                    [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
                    [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]])
```

2.1. Boucle ”for”, et opérations sur les vecteurs

2.1.1. Calculer la moyenne de la matrice en utilisant deux méthodes différentes.

2.2. Manipulation des pixels de l’image

2.2.1. Calculer la distance entre les points A, B et A, C on utilisant deux méthodes euclidienne et la distance de Manhattan

Soit le pixel p = f(5,3)

2.2.2. Donner la valeur du pixel p

2.2.3. Calculer la moyenne de (3*3) voisin de p (l'opérateur valeur moyenne est-il linéaire ?)

2.2.4. Calculer la médiane de (3*3) voisin de p (l'opérateur valeur médiane est-il linéaire?)

2.3. Manipulation des images

2.3.1. Lire l’image Im_01

2.3.2. Afficher l’image Im_01

2.3.3. Donner la hauteur et la largeur de Im_01

2.3.4. Comme bien de bits il nous faut pour sauvegarder cette image

• Soit l'image im_02 présentée par le tableau 2D suivant dans laquelle nous avons modifiés une petite partie à gauche en bas de Im_01

```
Im_02 =  np.array( [[255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 255, 255, 110, 110, 110, 110, 110, 110, 255, 255, 255],
                    [255, 255, 110, 110, 110, 110, 110, 110, 110, 110, 110, 110, 255],
                    [255, 255, 255, 170, 170, 170, 170, 170, 170, 170, 170, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255, 255, 255, 255, 170,  10,  170, 170,  10,  170, 255, 255, 255],
                    [255, 255, 255, 255, 170, 170, 170, 170, 170, 170, 255, 255, 255],
                    [255,   0,   0,   0,  255, 170,  80,  80,  170, 255, 255, 255, 255],
                    [255,   0,   0,   0,  255, 170, 170, 170, 170, 255, 255, 255, 255],
                    [255,   0,   0,   0,  255, 255, 255, 255, 255, 255, 255, 255, 255],
                    [255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255]])
```

2.3.5. Détecter cette modification en utilisant la soustraction entre Im_01 et Im_02 et donner l'image résultante.

2.3.6. Augmenter la luminance de l'image Im_01 en additionnant cette image avec elle-même

2.3.7. Exécuter la multiplication de l'image Im_01 avec un ratio pour améliorer le contracte