

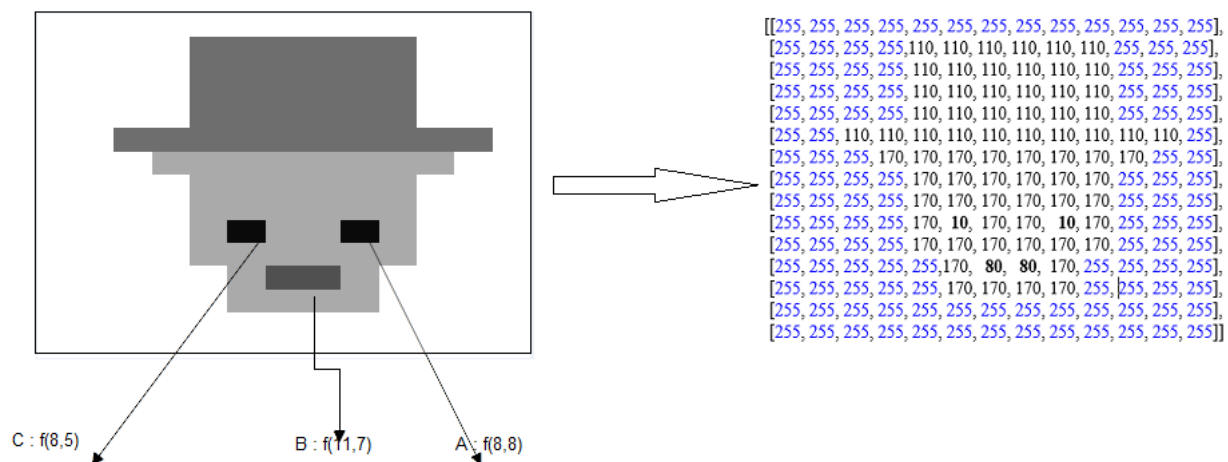
Practical Work N°1: Introduction to Image Processing with Python

Objectives

- Introduce the Python programming language.
- Introduce basic image processing commands in Python and become familiar with several image processing libraries, such as:
 - **NumPy**: supports multidimensional arrays.
 - **PIL / Pillow (Python Imaging Library)**: an open-source Python library that supports opening, manipulating, and saving many different image file formats.
 - **Scikit-image**: uses NumPy arrays as image objects and provides various algorithms for segmentation, color space manipulation, geometric transformations, analysis, morphology, feature detection, and more.
 - **SciPy**: another core scientific module in Python (like NumPy) used for basic image manipulation and processing tasks.
 - **OpenCV-Python**: OpenCV (Open Source Computer Vision Library) is one of the most widely used libraries for computer vision applications.
 - **Matplotlib**: specialized in 2D plotting of NumPy arrays as a cross-platform data visualization library.

Exercise Statement

Consider the following image and its grayscale matrix representation (Image 1).



Create the 2D array shown below

Answer:

```
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, 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]
])
```

1. For Loops and Vector Operations

- Calculate the mean of the matrix using two different methods.

2. Pixel Manipulation

- Calculate the distance between points A, B and points A, C using two distance metrics: Euclidean distance and Manhattan distance.
- Given the pixel $p = f(5, 3)$:
 - Provide the value of pixel p .
 - Calculate the mean of the 3x3 neighborhood of p . Is the mean operator linear?
 - Calculate the median of the 3x3 neighborhood of p . Is the median operator linear?

3. Image Processing Operations

- Read the image Im_01.
- Display the image Im_01.
- Provide the height and width of Im_01.
- How many bits are required to store this image?
- Consider the image Im_02 represented by the following 2D array (Table 1), in which a small region at the bottom-left of Im_01 has been modified:

```
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]
```

1)

- Detect this modification by subtracting Im_02 from Im_01 (or vice versa) and provide the resulting image.
- Increase the brightness/luminance of Im_01 by adding the image to itself.
- Perform scalar multiplication on Im_01 with a scaling ratio to improve contrast.