



Ecole Supérieure Informatique Sidi Bel Abbès
Second Cycle
Computer Vision
Lab -1-: Point Processing in Image Processing

October 29, 2024

1. Objective:

- To understand the basic concepts of point processing in image processing.
- To implement simple point processing operations on images.
- To observe the effects of various image processing techniques on images

2. Programming Language and Library

- Python .
- matplotlib library.
- Jupyter Notebook or a code editor for Python

3. Lab Procedure:

- Select an image from your directory.
- Load the image using matplotlib.

```
import numpy as np
from matplotlib import pyplot as plt
```

```
img = plt.imread('images/flowers.png')
plt.imshow(img)
plt.title('image')
plt.axis('off')
plt.show()
```

- NumPy for Images
 - Matplotlib imread under Python returns a NumPy array,
 - The shape attribute keeps the array's dimensions,
 - A RGB color image can be represented by a $M \times N \times 3$ uint8 array,
 - The color triplet can be retrieved indexing the pixel position,
 - Indexing can be used to retrieve a specific channel.

```
print(type(im))
print(im.shape)
print(im.ndim)
print(im.size)
print(im.dtype)
```

- Print the shape of the grayscale image
 - Print the shape of a grayscale image

- Slicing
 - Slicing can retrieve parts of an array,
 - Employs the convention start:stop:step.

```
plt.imshow(img[100:400:,100:400,:])
plt.show()
```

- RGB Channels
 - RGB channels are the essential components that collectively create the full spectrum of colors in digital images.

```
R = img[:, :, 0]
G = img[:, :, 1]
B = img[:, :, 2]

output = [img, R, G, B]
titles = ['Original Image', 'R Channel', 'G Channel', 'B Channel']

for i in range(4):
    plt.subplot(2, 2, i+1)
    plt.axis('off')
    plt.title(titles[i])
    if i == 0:
        plt.imshow(output[i])
    else:
        plt.imshow(output[i], cmap='gray')

plt.show()
```

- Stacking them together

```
Output = np.dstack((R, G, B))
plt.imshow(Output)
plt.show()
```

- Motion Difference
 - Load and plot the two images named (t4,t5) using matplotlib,
 - Subtract the pixel values of **t4.png** from **t5.png** to create a new image.
 - Create a figure with a subplot to display the resulting image.
- Apply various point processing techniques to the loaded image.
 - Making the image overall darker,
 - Reduce the difference between light and dark areas,
 - Change all the pixel values from dark to light and vice versa,
 - Increase the brightness of the pixels, making the image overall brighter,
 - Increasing the difference between light and dark areas.

4. **Homework Assignment:** Provide a report on the outcomes, including the code used and the purpose of applying these techniques to the chosen images.