



University of Science and Technology of Hanoi  
Address: USTH, 18 Hoang Quoc Viet, Cau Giay, Hanoi  
Telephone/ Fax: +84-4 37 91 69 60  
Email: [officeusth@usth.edu.vn](mailto:officeusth@usth.edu.vn)  
Website: <http://www.usth.edu.vn>

## IMAGE PROCESSING

### Lab Session 1: Basic Image Processing with OpenCV and Python

#### Part 1. Install OpenCV with Python:

- Install OpenCV with Python on your computer.
- Practice how to execute a code file written in Python:

```
$ python file_name.py
```

#### Part 2. Download from the Internet a color image, then do the following image processing tasks using OpenCV and Python:

**Exercise 1:** Read the downloaded image using the function `cv2.imread()`, then display the image using the matplotlib function `imshow()`

**Exercise 2:** Resize the downloaded image using the function `cv2.resize()`

**Exercise 3:** Convert the colour image to grayscale image, using the following formula:

$$img_{grey} = 0.2126 R + 0.7152 G + 0.0722 B$$

**Exercise 4:** Change brightness of the downloaded image, using the following formula:

$$img_{processed} = a * f(x, y) + b$$

In which,  $f(x, y)$  is the original image ( $img$ ) at the coordinate  $(x, y)$ ;  $a$  and  $b$  are user-defined constants.

**Exercise 5:** Image binarization:

- Convert the downloaded colour image to grayscale image.
- Apply the global thresholding technique to binarize the image.
- Apply the adaptive thresholding technique to binarize the image, using the method: `cv2.ADAPTIVE_THRESH_MEAN_C`, `blockSize = 25`, `constant C = 12`

**Exercise 6:** Histogram equalization:

- Convert the colour image to grayscale image, display the original image.
- Calculate and display the histogram of original image, using the function `cv2.calcHist()`
- Use the function `cv2.equalizeHist()` to calculate the new equalized image.
- Display the equalized histogram.

**Exercise 7:** Image Filtering

- Apply the Average filter using the function: `cv2.blur()`, Gaussian filter using the function `cv2.GaussianBlur()` , Median filter using the function `cv2.medianBlur()` on the downloaded images to blur the image.
- Display the original images and the filtered images to see the difference.

**Exercise 8:** Edge detection

- Apply Laplacian filter using the function `cv2.Laplacian()`, Sobel filter using the function `cv2.Sobel()` to detect edges of the downloaded image.
- Display the original images and the filtered images to see the difference.

**Notice:** you are required to upload the codes of your labworks to the google drive folder of the DIP course.