

# THỰC HÀNH BUỔI 2

## Xử lý tổ chức đồ. Tích chập

### MỤC TIÊU

Viết chương trình bằng Python trên Google Colab thực hiện các thao tác xử lý tổ chức đồ và lọc ảnh dùng phép tích chập.

### NỘI DUNG

#### Phần 1: Xử lý tổ chức đồ

#### Phần 2: Phép tích chập

### TÀI LIỆU THAM KHẢO

Hướng dẫn lập trình ngôn ngữ python  
[www.cit.ctu.edu.vn/~dtnghe/oss/python.pdf](http://www.cit.ctu.edu.vn/~dtnghe/oss/python.pdf)

#### Phần 1: Xử lý tổ chức đồ

**Bước 1.** Làm bài TH1 nếu chưa làm.

**Bước 2.** Tạo mới notebook trên Google Colab, đặt tên là XLA\_TH2.

**Bước 3.** Tải một trong số các ảnh đã có vào mảng **img\_ORG**, tách 3 kênh màu vào 3 mảng **img\_ORG\_R**, **img\_ORG\_G** và **img\_ORG\_B** và hiển thị.

**Bước 4.** Ghép 3 kênh màu lại thành ảnh ban đầu bằng lệnh sau và hiển thị:

```
output1 = np.dstack((img_ORG_R, img_ORG_G, img_ORG_B))
```

**Bước 5.** Khai báo hàm XLA\_linear() trong khối lệnh mới như sau và chạy:

```
def XLA_linear(img, a =1, b=0):  
    kq = img.astype('float32')  
    kq = kq*a + b  
    kq = kq.clip(0,255)  
    return kq.astype('uint8')
```

**Bước 6.** Khai báo hàm XLA\_to\_grayscale() trong khối lệnh mới như sau và chạy:

```

def XLA_to_grayscale(img):
    R = img_ORG[:, :, 0]
    G = img_ORG[:, :, 1]
    B = img_ORG[:, :, 2]
    return (R* 0.2989 + G*0.5870 + B*0.1140).clip(0, 255).astype('uint8')

```

**Bước 7:** Thực hiện trượt tổ chức đồ ảnh gốc và hiển thị:

```

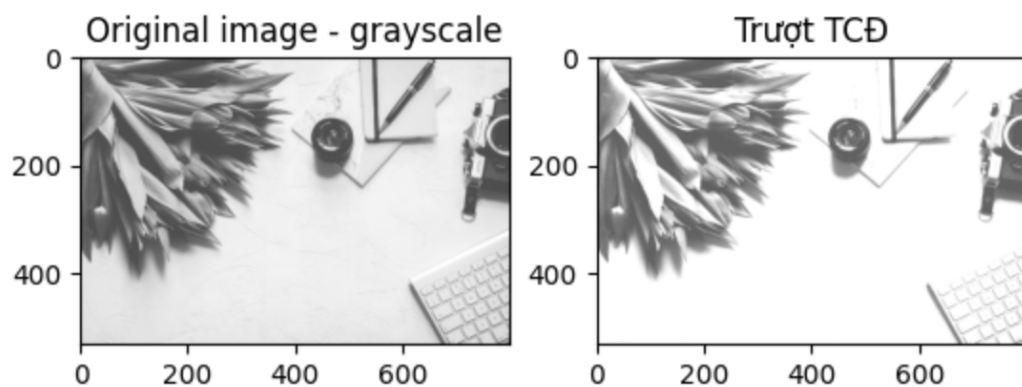
plt.subplots(nrows = 1, ncols =2)

plt.subplot(1,2,1)
plt.title('Original image - grayscale')
output2 = XLA_to_grayscale(img_ORG)
plt.imshow(output2, cmap='gray')

plt.subplot(1,2,2)
plt.title("Trượt TCD")
plt.imshow(XLA_linear(output2, 1, 50), cmap='gray')

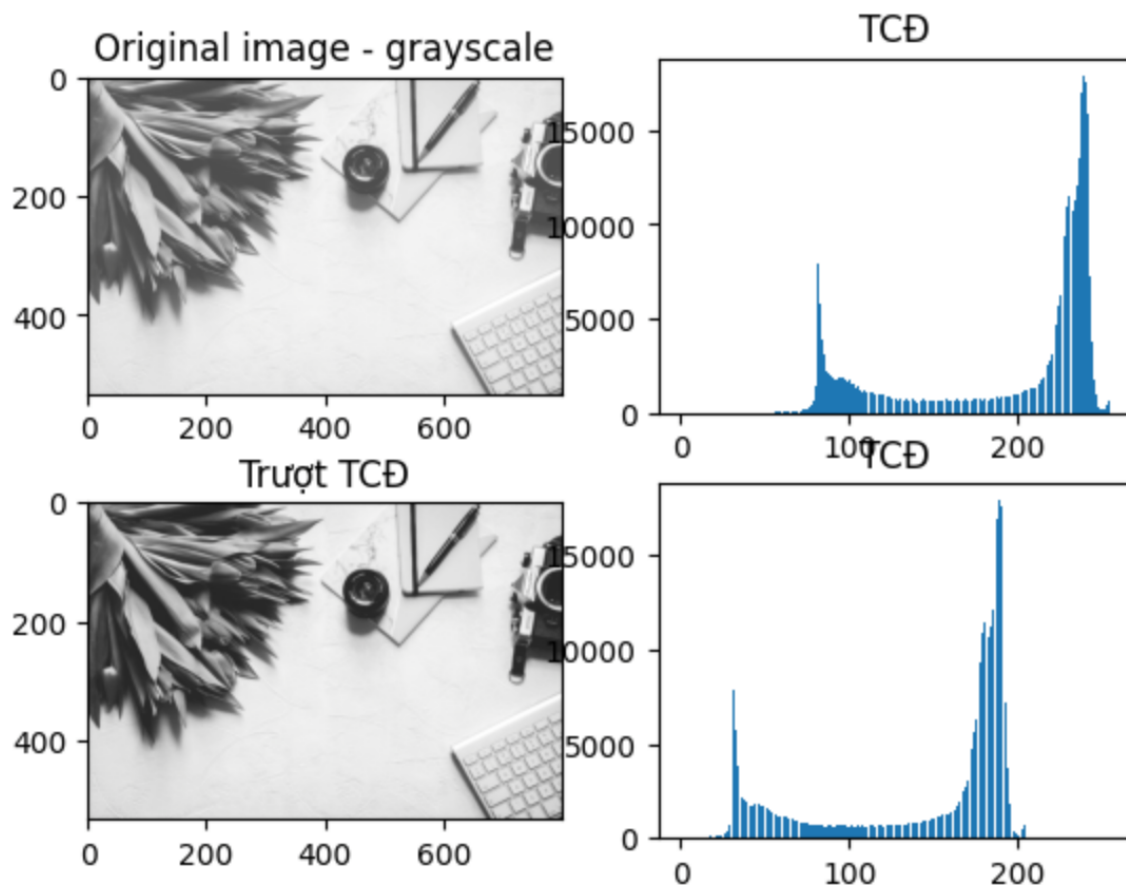
```

 <matplotlib.image.AxesImage at 0x7be63d5ef190>



**Bài tập:**

- Viết lệnh vào khối lệnh tiếp theo để thu được kết quả sau:



2. Tương tự câu 1, nhưng cần căng tổ chức đồ, sao cho điểm ảnh tối nhất có giá trị 0, điểm sáng nhất – 255.

## Phần 2: Phép tích chập

**Bước 1:** Chạy khối lệnh sau để nạp các hàm cần thiết:

```
from scipy import signal
from skimage.filters import gaussian
```

**Bước 2:** Hàm `signal.convolve2d` thực hiện phép tích chập trên mảng 2 chiều. Chạy khối lệnh sau:

```

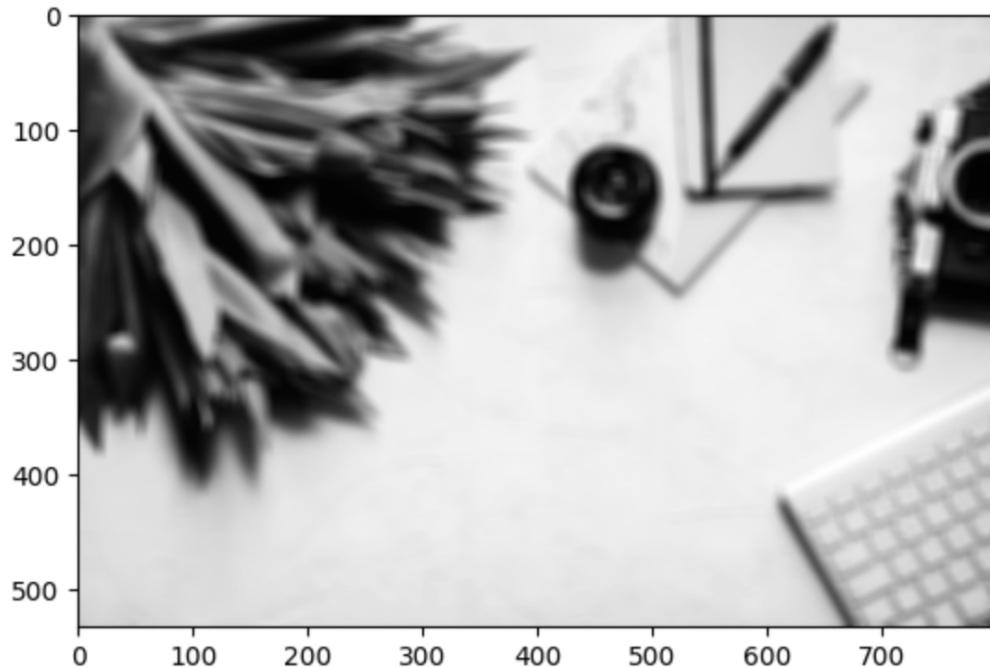
kernel = np.ones( (11,11), np.float32)/121
img3 = signal.convolve2d(output2, kernel, boundary = 'symm', mode = 'same')
plt.imshow(img3, cmap = 'gray')
plt.show

```

```

<function matplotlib.pyplot.show(close=None, block=None)>

```



### Bài tập:

- Viết lệnh tính các nhân khác đã học, thực hiện phép tích chập và hiển thị ảnh kết quả.
- So sánh ảnh kết quả khi thực hiện tích chập với nhân Gaussian vừa thực hiện, với kết quả thực hiện lời gọi hàm `gaussian(output3, sigma = 3)`

### \*\*\*Nộp bài:

- Chia sẻ thư mục Colab notebooks cho tài khoản [tvchau@ctu.edu.vn](mailto:tvchau@ctu.edu.vn) quyền "Xem";
- Download notebook ở cả 2 dạng .py và .ipynb, đặt tên dạng XLA\_TH2\_MãSV\_Họ\_và\_tên.\*
- Nộp 2 file này qua Elearning.ctu.edu.vn