

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC

| Đề tài |

IMAGE PROCESSING WITH NUMPY

| Giảng viên hướng dẫn |

GV. Phan Thị Phương Uyên

Môn học: Toán ứng dụng và thống kê



Võ Trần Quang Tuấn - 18127248

Thành phố Hồ Chí Minh – 2020

I. Sinh viên thực hiện.

- Họ và tên: Võ Trần Quang Tuấn.
- MSSV: 18127248
- Lớp: 18CLC2
- Môn học: Toán ứng dụng và thống kê.

II. Bài toán giải quyết.

1. Mô tả.

- Xử lý ảnh có vai trò quan trọng trong các hệ thống thông minh như camera hành trình, xe tự hành, đặc biệt là robot có tích hợp các chương trình và thiết bị thị giác thông minh.
- Ngoài ra xử lý ảnh giúp các ảnh số thêm sắc nét hơn, có vai trò quan trọng trong việc rút trích đặc trưng ứng dụng trong các bài toán học máy.

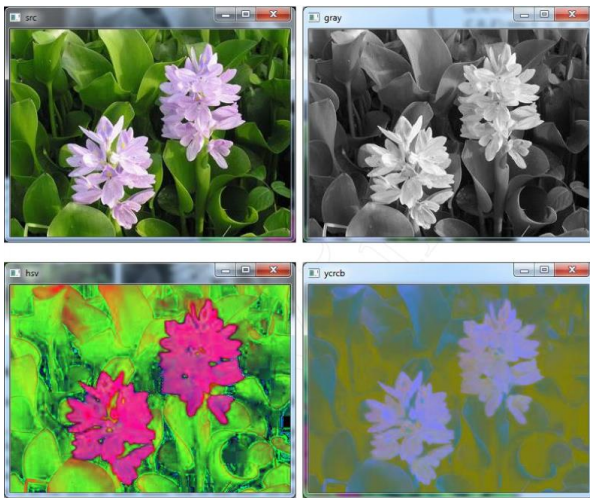


Figure 1: Xử lý ảnh (source: internet).

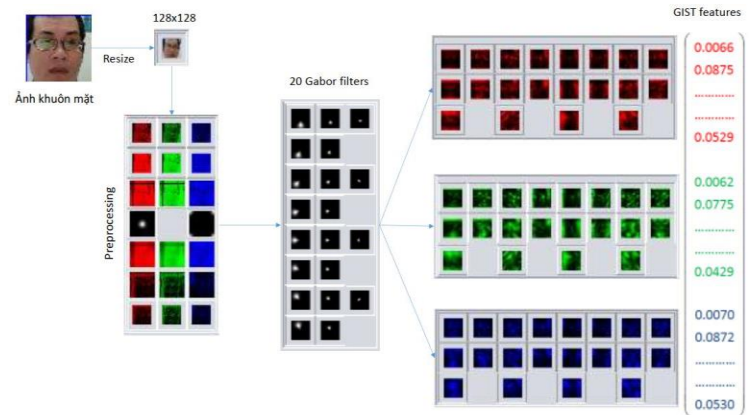


Figure 2: Rút trích đặc trưng ảnh (source: internet).

- Trong đồ án này, chúng ta sẽ giải quyết một số công việc trong xử lý ảnh đơn giản nhưng không kém phần quan trọng, đó là: thay đổi độ sáng, thay đổi độ tương phản, chuyển ảnh xám, thay đổi chiều, chồng hai ảnh thành một, làm mờ ảnh.

2. Hướng giải quyết và ý tưởng.

a. Thay đổi độ sáng và độ tương phản của ảnh.

- Để thay đổi độ sáng của ảnh, cộng mỗi giá trị trong mỗi channel của mỗi pixel của ảnh thêm một giá trị bất kỳ.
- Để thay đổi độ tương phản của ảnh, nhân mỗi giá trị trong mỗi channel của mỗi pixel của ảnh thêm một giá trị bất kỳ.

b. Chuyển ảnh thành ảnh xám.

- Công thức chuyển ảnh xám:

$$S = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$$

Với R, G, B là giá trị 3 channel của mỗi pixel.

- Phương pháp: Tính giá trị S cho mỗi pixel trong ảnh, xếp lại thành một ma trận với shape bằng shape của ảnh gốc, ta được ma trận ảnh xám.

c. Thay đổi chiều của ảnh (lật ảnh ngang – dọc).

- Lật ngang: lấy đối xứng ma trận ảnh qua chiều cao ảnh.
- Lật dọc: lấy đối xứng ma trận ảnh qua chiều dài ảnh.

d. Chồng hai ảnh xám có cùng kích thước.

- Đầu tiên, hai ảnh đầu vào phải được chuyển qua ảnh xám, sau đó sử dụng công thức:

$$S = 0.5 \times image01 + 0.5 \times image02$$

- Sau đó, chúng ta sẽ được ma trận S chính là ma trận ảnh sau khi chồng.

e. Làm mờ ảnh.

- Chúng ta sẽ dùng một ma trận để làm filter (hay còn gọi là kernel) để quét qua ảnh và tính tích chập với vùng được quét qua.
- Phép toán tích chập (convolution) thường được dùng nhiều trong các mô hình CNN trong việc rút trích đặc trưng và huấn luyện mô hình.

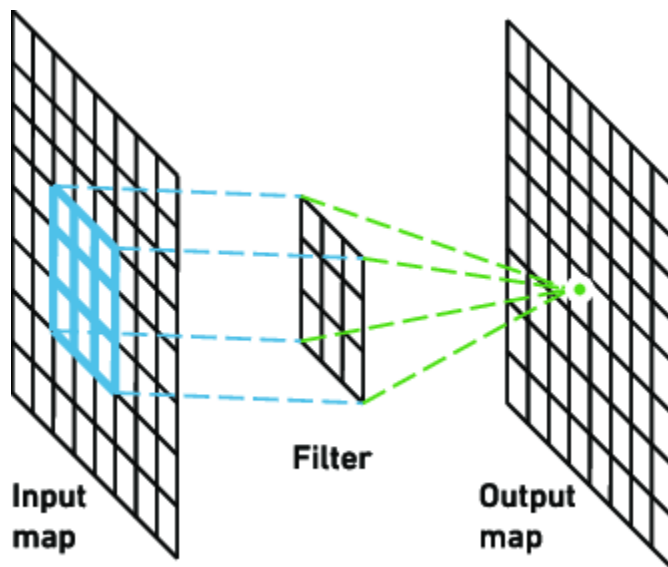


Figure 3: Phép toán tích chập (convolution) - source: internet.

- Nguyên tắc tính tích chập: nhân element-wise các phần tử của filter với vùng nó quét qua và lấy tổng, ta được một giá trị mới. Giá trị này đưa vào một ma trận mới theo thứ tự tính tích chập lúc này, ma trận này gọi là feature map.

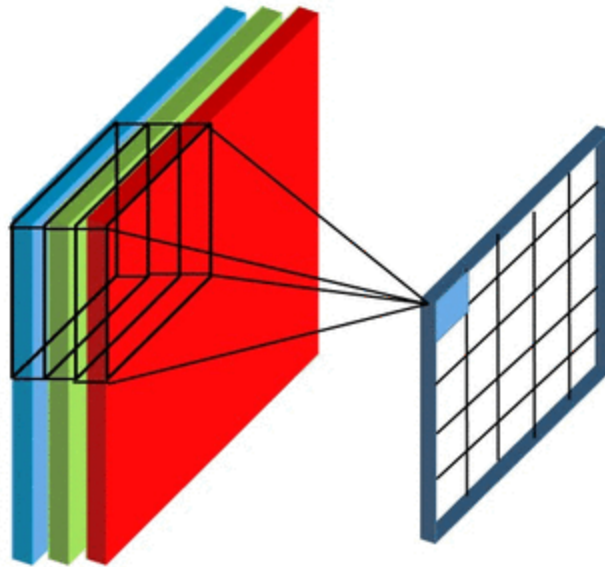


Figure 4: Convolution with RGB channel.

- Sau khi tính tích chập, feature map sẽ nhỏ hơn kích thước ban đầu, nói cách khác, chính là output của chúng ta sẽ bị giảm đi kích thước, cho nên để giữ nguyên kích thước output, người ta thường padding ma trận ảnh gốc trước khi tính tích chập, tức là tạo viền toàn số 0 cho ảnh.

0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0


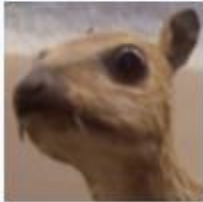
Kernel		
0	-1	0
-1	5	-1
0	-1	0

114				

Figure 5: Padding image (source: internet).

- Ngoài ra chúng ta còn có một tham số liên quan việc tính tích chập, đó là stride, stride là số bước nhảy khi quét ma trận ảnh của filter.

- Trong việc làm mờ ảnh, chúng ta sẽ sử dụng filter có tên Box Blur hoặc Gaussian Blur.

Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

- Tùy theo kích cỡ filter và ảnh mà chúng ta có stride với padding khác nhau, trong trường hợp làm mờ ảnh, chúng ta sử dụng padding = 1 (tạo viền 0 một lần), stride = 1 (bước nhảy là 1).
- Tách ma trận ảnh gốc thành 3 ma trận tương ứng 3 kênh màu, tính tích chập cho từng kênh với filter như trên ta được 3 feature map, sau đó xếp lại thành một ma trận ảnh mới theo thứ tự RGB, ta được ma trận ảnh gốc đã làm mờ.

III. Các chức năng đã hoàn thành và mô tả hàm.

1. Các chức năng.

- Thay đổi độ sáng của ảnh.
- Thay đổi độ tương phản của ảnh.
- Chuyển ảnh thành ảnh xám.
- Thay đổi chiều của ảnh (lật ảnh ngang - dọc).
- Chồng hai ảnh xám có cùng kích thước.
- Làm mờ ảnh.

2. Các thư viện sử dụng.

- Thư viện dùng cho tính toán ma trận: **numpy**
- Thư viện dùng để đọc ảnh và mở ảnh: **PIL**
- Thư viện dùng để đọc ma trận ảnh và show ảnh: **matplotlib.pyplot**

3. Các hàm sử dụng.

a. Thay đổi độ sáng của ảnh.

`change_brightness(image_name, brightness_lv=0)`

➔ Input: tên ảnh, độ sáng.

➔ Output: ảnh được thay đổi độ sáng.

b. Thay đổi độ tương phản của ảnh.

`change_contrast(image_name, contrast_lv=1)`

- Input: tên ảnh, độ tương phản.
- Output: ảnh được thay đổi độ tương phản.

c. **Chuyển ảnh thành ảnh xám.**

`grayscale_image(image_name)`

- Input: tên ảnh
- Output: ảnh được chuyển thành ảnh xám.

d. **Thay đổi chiều của ảnh (lật ảnh ngang - dọc).**

`rotate_image(image_name, method='horizontal')`

- Input: tên ảnh
- Output: ảnh được lật ngang nếu method = 'horizontal', lật dọc nếu method = 'vertical'

e. **Chồng hai ảnh xám cùng kích thước.**

`blend_2_gray_image(image_name_01, image_name_02)`

- Input: tên hai ảnh cần chồng lên.
- Output: hai ảnh đã được chồng lên nhau và biến thành màu xám.

f. **Làm mờ ảnh.**

`convolve(conv_unit, kernel)`

- Input: vùng ảnh được quét, kernel
- Output: giá trị tích chập

`make_convolution_matrix(image_padding, kernel_shape)`

- Input: ảnh được padding, kích thước kernel
- Output: ma trận chứa các vùng được quét bởi kernel

`convolution(image, stride=1, padding=1, method='Box Blur')`

- Input: ma trận ảnh, loại kernel
 - Output: ma trận tích chập.
- `blurring_image(image_name, method='Box Blur')`
- Input: tên ảnh, loại kernel
 - Output: ma trận ảnh đã được làm mờ.

IV. **Kết quả thu được.**

1. **Kết quả.**

a. **Thay đổi độ sáng.**



Figure 6: ảnh gốc.



Figure 7: ảnh sau khi thay đổi độ sáng.

b. Thay đổi độ tương phản.



Figure 8: ảnh gốc.



Figure 9: ảnh sau khi thay đổi độ tương phản.

c. Chuyển ảnh thành ảnh xám.



Figure 10: ảnh gốc.



Figure 11: ảnh xám.

d. Thay đổi chiều của ảnh (lật ngang – dọc).



Figure 12: ảnh gốc.



Figure 13: ảnh lật ngang.



Figure 14: ảnh gốc.



Figure 15: ảnh lật dọc.

e. Chồng hai ảnh xám có cùng kích thước.

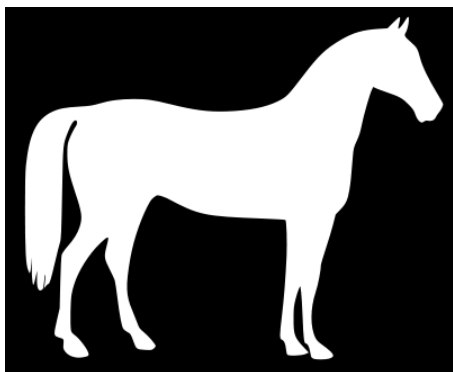


Figure 16: ảnh gốc 1



Figure 17: ảnh gốc 2.



Figure 18: ảnh xám sau khi chồng ảnh 1 và 2

f. Làm mờ ảnh.



Figure 19: ảnh gốc.



Figure 20: ảnh sau khi làm mờ.

2. Nhận xét.

- Các phương pháp xử lý ảnh (ngoài làm mờ), thời gian chạy nhanh vì ít các phép nhân hơn so với làm mờ ảnh nếu chưa chuẩn hoá ảnh.
- Phương pháp làm mờ ảnh bằng cách tính tích chập, nếu chưa chuẩn hoá ảnh (chia cho 255, hoặc đưa trung bình về 0 và độ lệch chuẩn về 1,...) thì quá trình tính tích chập diễn ra khá lâu đối với input lớn, cho nên muốn cải tiến thì cần chuẩn hoá ảnh trước khi làm mờ.

V. Tài liệu tham khảo.

[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))

https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm