

Report Lab03 – Image Processing

AUGUST 12

Toán Ứng dụng và Thống Kê

Tác giả: Phạm Ngọc Thùy Trang – 18127022

PROJECT 02 – LAB03

THÔNG TIN CÁ NHÂN

Họ và tên: Phạm Ngọc Thùy Trang

MSSV: 18127022

Lớp: 18CLC1

GIẢNG VIÊN HƯỚNG DẪN

GV. Phan Thị Phương Uyên

GV. Nguyễn Văn Quang Huy

Ý TƯỞNG THỰC HIỆN CÁC HÀM CHỨC NĂNG

1. Giới thiệu và mô tả khái quát xử lý ảnh là gì?

Xử lý ảnh là dùng các thuật toán thay đổi hình ảnh đầu vào để tạo ra hình ảnh mới, lần đầu vào và đầu ra đều là ảnh. Nó có ý nghĩa rất quan trọng trong cuộc sống không chỉ là đối với các hệ thống thông minh như xe tự vận hành, robot tích hợp các chương trình và thiết bị thị giác thông minh

Bên cạnh đó, các tính năng như làm giảm độ nhiễu ảnh, tìm góc cạnh, xử lý tăng/giảm độ tương phản, tăng/giảm độ sáng của ảnh, chuyển ảnh sang màu xám, chồng hai ảnh cùng kích thước, v.v... đồng thời cũng là nhiệm vụ của đồ án này

2. Ý tưởng thực hiện các chức năng

a. Thay đổi độ sáng của ảnh

Để thay đổi độ sáng của ảnh ta cộng mỗi giá trị trong mỗi kênh của mỗi điểm ảnh của ảnh thêm một giá trị bất kỳ.

b. Thay đổi độ tương phản của ảnh

Để thay đổi độ tương phản của ảnh ta nhân mỗi giá trị trong mỗi kênh của mỗi điểm ảnh của ảnh thêm một giá trị bất kỳ.

c. Chuyển ảnh thành ảnh xám

Với R,G,B là 3 giá trị cho 3 kênh của mỗi điểm ảnh, ta có công thức để chuyển ảnh sang ảnh xám như sau:

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

Bằng cách tính giá trị N cho mỗi điểm ảnh trong ảnh gốc, ta sẽ xếp những điểm ảnh mới này lại thành một ma trận với shape bằng shape của ảnh gốc thì ta được ma trận ảnh xám → ảnh chuyển sang ảnh xám.

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

- **Lật ngang:** Bằng cách lấy đối xứng ma trận ảnh qua chiều cao ảnh, ta được một ảnh được lật ngang.
- **Lật dọc:** Bằng cách lấy đối xứng ma trận ảnh qua chiều dài ảnh, ta được một ảnh lật dọc.

e. Làm mờ ảnh

- Ta sẽ tạo ra một Gaussian blur - một loại bộ lọc làm mờ ảnh bằng cách sử dụng hàm Gauss để tính toán sự biến đổi áp dụng lên cho từng điểm ảnh của ảnh đó. Ở đây ta dùng Gaussian Blur 3x3
- 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.
- Bên cạnh đó ta cũng sẽ dùng phép tính chập (convolution) (thường được dùng trong các mô hình CNN) để rút trích đặc trưng và huấn luyện mô hình.
- Sau khi convolution thì do feature map sẽ có kích thước nhỏ hơn kích thước ban đầu, nên để có thể xuất ra một ảnh không bị giảm đi kích thước, ta sẽ dùng padding ma trận ảnh gốc trước khi tính tích chập, tức là tạo viền toàn số 0 cho toàn ảnh.
- Cuối cùng, ta sẽ tách ma trận ảnh gốc thành 3 ma trận tương ứng 3 kênh màu và tính tích chập cho từng kênh với bộ lọc như trên ta được 3 feature map, xếp lại thành một ma trận ảnh mới theo thứ tự RGB, ta được ma trận ảnh gốc đã làm mờ.

f. Chồng hai ảnh có cùng kích thước

Trước khi chồng ảnh, ta cần phải resize chúng về cùng kích thước, sau đó mới dùng hàm chuyển ảnh sang ảnh xám ở trên cho hai ảnh đầu vào và cuối cùng là dùng công thức:

$$N = 0.5 \times \text{img01} + 0.5 \times \text{img02}$$

Thì khi đó ta sẽ được ma trận N chính là ma trận ảnh sau khi chồng hai ảnh lại với nhau

CÁC CHỨC NĂNG ĐÃ HOÀN THÀNH

Tên chức năng	Đánh giá tỉ lệ hoàn thành
Thay đổi độ sáng của ảnh	1/1
Thay đổi độ tương phản của ảnh	1/1
Chuyển ảnh thành ảnh xám	2/2
Thay đổi chiều của ảnh gồm: +Lật dọc +Lật ngang	2/2
Chồng hai ảnh có cùng kích thước	2/2
Blur (làm mờ) ảnh	2/2
Tổng:	10/10

CÁC THƯ VIỆN SỬ DỤNG:

Thư viện dùng cho tính toán: **numpy**. Được phép sử dụng thoải mái thư viện này.

Thư viện chỉ được dùng để đọc ảnh và mở ảnh : **PIL**

Thư viện chỉ được dùng để đọc ma trận ảnh và imshow ảnh: **matplotlib**

MÔ TẢ CÁC CHỨC NĂNG

(Ở đây ta viết ở từng cell là mỗi một chức năng nên sẽ không viết hàm, chỉ có một hai chức năng sẽ viết hàm. Chính vì vậy ta sẽ mô tả các chức năng thông qua minh họa bằng code và bằng lời)

Tên chức năng	Minh họa bằng code	Mô tả chính
Thay đổi độ sáng của ảnh	<pre>beta = 50 new_image = np.zeros(image.shape, image.dtype) for y in range(image.shape[0]): for x in range(image.shape[1]): for c in range(image.shape[2]): new_image[y,x,c] = np.clip(image[y,x,c] + beta, 0, 255) matplotlib.pyplot.imshow(new_image)</pre>	<p>Tăng giá trị pixel đồng đều trên tất cả các kênh cho toàn bộ hình ảnh để tăng độ sáng hoặc giảm giá trị pixel đồng đều trên tất cả các kênh để toàn bộ hình ảnh giảm độ sáng.</p> <p>Tăng độ sáng của ảnh làm cho hình ảnh trắng hơn, ở đây ta khai báo một con số beta để thể hiện độ sáng của ảnh, nếu beta càng lớn thì ảnh</p>

		<p>càng sáng, đến một lúc nào đó thì ảnh gần như bị cháy nắng. Ta không thay đổi độ sáng của ảnh dựa trên kênh HSL, ta chỉ làm trên kênh RGB nên sẽ khởi tạo beta bằng một con số vừa phải. Lưu ý rằng nếu giá trị điểm ảnh nào vượt quá 255 sẽ được đưa về giá trị 255, giá trị điểm ảnh nào dưới giá trị 0 sẽ được đưa về 0.</p>
Thay đổi độ tương phản của ảnh	<pre>alpha = 1.5 #giá trị alpha để điều chỉnh độ tương phản, nếu giá trị new_image = np.zeros(image.shape, image.dtype) for y in range(image.shape[0]): for x in range(image.shape[1]): for c in range(image.shape[2]): new_image[y,x,c] = np.clip(alpha*image[y,x,c], 0, 255) matplotlib.pyplot.imshow(new_image)</pre>	<p>Ta cũng khởi tạo giá trị alpha để điều chỉnh độ tương phản, nếu giá trị alpha = 1.0 thì ảnh bình thường. Lưu ý rằng nếu giá trị</p>

		điểm ảnh nào vượt quá 255 sẽ được đưa về giá trị 255, giá trị điểm ảnh nào dưới giá trị 0 sẽ được đưa về 0.
Chuyển ảnh thành ảnh xám	<pre>def convert_image2_grayscale(image):</pre>	Đầu vào là một image và đầu ra là một ảnh đã được convert sang ảnh xám bằng cách áp dụng công thức ở phần mô tả ý tưởng để thay đổi giá trị điểm ảnh
Thay đổi chiều của ảnh gồm: +Lật dọc +Lật ngang	<pre>#4. Lật ảnh (ngang - dọc) flipimage = np.fliplr(image) matplotlib.pyplot.imshow(flipimage)</pre> <pre>flipimage = np.flipud(image) matplotlib.pyplot.imshow(flipimage)</pre>	<p>Bằng cách dùng hàm flipup và fliplr của thư viện numpy để lật ảnh dọc và lật ảnh ngang, với đầu vào là ảnh gốc.</p> <p>Hàm flipup của numpy được dùng để lật mảng lên hoặc xuống bằng</p>



		<p>cách lật các entry ở mỗi cột theo chiều hướng lên/xuống. Các hàng thì được giữ nguyên, nhưng xuất hiện với thứ tự khác so với ảnh gốc.</p> <p>Tương tự với hàm <code>flipr</code> cũng được dùng để lật ảnh sang trái/phải bằng cách lật các entry của mỗi hàng theo chiều hướng sang trái/phải. Các cột sẽ được giữ nguyên, nhưng chúng cũng sẽ được sắp xếp theo thứ tự khác với thứ tự cũ.</p>
<p>Chồng hai ảnh có cùng kích thước</p>	<pre>def blend_image(image1, image2, threshold=0.5):</pre>	<p>Tham số truyền vào là hai ảnh bất kỳ và một giá trị 0.5. Giá trị này</p>

		<p>được dùng để khi hai ảnh đầu vào sau khi được resize và chuyển sang ảnh xám sẽ tạo ra hai ma trận ảnh mới. Hai ma trận ảnh mới đó sẽ nhân với 0.5 để tạo ra ma trận ảnh có hai ảnh bị chồng lên nhau.</p>
Blur (làm mờ) ảnh	<pre>def gaussianFormular(x,y,sigma):</pre>	<p>Hàm công thức của Gaussian, với x là khoảng cách từ điểm gốc cho đến trục hoành, y là khoảng cách từ điểm gốc cho đến trục tung và sigma là độ lệch chuẩn của phân phối Gauss.</p>
	<pre>def gaussFilter(size, sigma):</pre>	<p>Với mỗi phép lọc ta có những ma trận lọc (Kernel) khác nhau, và kích</p>

		<p>thước ma trận M này thường là một số lẻ. Đầu vào: Size ở đây là kích thước ma trận lọc (có thể là 3×3, 5×5, ...). Lưu ý rằng độ lớn của ma trận lọc cần được lựa chọn cho đủ rộng. Tham số sigma sẽ quyết định tác dụng của bộ lọc Gauss lên ảnh.</p>
	<pre>def convolution(image, filter):</pre>	<p>Đầu vào là bức ảnh và bộ lọc mà ta áp dụng từ hàm tính toán <code>gaussFilter</code> ở trên. Bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss, sau đó cộng chúng lại để tạo thành ảnh đầu ra.</p>

KẾT QUẢ - HÌNH ẢNH TƯƠNG ỨNG VỚI TỪNG CHỨC NĂNG

Tên chức năng	Đầu vào	Đầu ra
Thay đổi độ sáng cho ảnh		
Điều chỉnh độ tương phản của ảnh		

<p>Chuyển đổi ảnh màu sang ảnh xám</p>		
<p>Lật ảnh ngang</p>		
<p>Lật ảnh dọc (1)</p>		

<p>Làm mờ ảnh sử dụng Gaussian filter</p>		
<p>Chồng hai ảnh có cùng kích thước: <i>chỉ làm trên ảnh xám</i></p>	<p>ẢNH 1</p> 	
	<p>ẢNH 2</p> 	

REFERENCES

[1]: https://en.wikipedia.org/wiki/Gaussian_blur

[2]:
https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.html

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