

**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 2

Image Processing

**MÔN: TOÁN ỨNG DỤNG VÀ THỐNG KÊ CHO CÔNG NGHỆ
THÔNG TIN**

**SINH VIÊN
MSSV
LỚP
KHOA**

**: NGUYỄN HOÀI MÃN
: 20127561
: 20CLC05
: CÔNG NGHỆ THÔNG TIN**

Ngày 05 tháng 07 năm 2022

MỤC LỤC

I. Các chức năng đã hoàn thành:	1
II. Ý tưởng thực hiện và mô tả về hàm chức năng:	1
1. Hàm Brightness(Thay đổi độ sáng cho ảnh)	1
2. Hàm Contrast(Thay đổi độ tương phản)	1
3. Hàm FlipDown - FlipLeft (Lật ảnh dọc - ngang)	2
4. Hàm Grayscale1-Grayscale2 (Chuyển đổi ảnh RGB thành ảnh xám)	2
5. Hàm Stack2image (Chồng 2 ảnh cùng kích thước: chỉ làm trên ảnh xám)	2
6. Blur (Làm mờ ảnh)	2
7. Hàm Circle (Cắt nội dung ảnh theo khung được áp lên (hình tròn))	3
8. Hàm Elip (Cắt nội dung ảnh theo khung được áp lên (hình elip))	3
III. Hình ảnh kết quả với từng chức năng:	4
IV. Tài liệu tham khảo	8

I. Các chức năng đã hoàn thành:

STT	Chức năng	Tỉ lệ hoàn thành(%)
1	Thay đổi độ sáng cho ảnh	100
2	Thay đổi độ tương phản	100
3	Lật ảnh (ngang- dọc)	100
4	Chuyển đổi ảnh RGB thành ảnh xám	100
5	Chồng 2 ảnh cùng kích thước: chỉ làm trên ảnh xám	100
6	Làm mờ ảnh	100
7	Cắt nội dung ảnh theo khung được áp lên (hình tròn)	100
8	Cắt nội dung ảnh theo khung được áp lên (hình elip)	50
9	Hàm main	100

II. Ý tưởng thực hiện và mô tả về hàm chức năng:

1. Hàm Brightness(Thay đổi độ sáng cho ảnh)

- Ý tưởng: tăng giá trị RGB của các điểm ảnh bằng cách cộng với một số nguyên.
- Input: đầu vào sẽ là một ảnh(img) và một số nguyên(bright)
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img)
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó ta sẽ xét từng điểm ảnh trên từng hàng từng cột lấy ra các giá trị R,B,G và cộng một số nguyên(bright) vào mỗi R,B,G của điểm ảnh đó. Giá trị của các điểm ảnh đó sẽ được cài đặt để không lớn hơn 255 và nhỏ hơn 0, nếu lớn hơn 255 thì sẽ cài lại mặc định là 255 còn nhỏ hơn 0 thì sẽ mặc định là 0.

2. Hàm Contrast(Thay đổi độ tương phản)

- Ý tưởng: tăng giá trị RGB của các điểm ảnh bằng cách nhân với một số factor, factor sẽ được tính theo công thức sau:
$$\text{factor} = (259 * (\text{contrast} + 255)) / (255 * (259 - \text{contrast}))$$
- Input: đầu vào sẽ là một ảnh(img) và một số nguyên(contrast).
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó ta sẽ xét từng điểm ảnh trên từng hàng từng cột lấy ra các giá trị R,B,G và nhân với factor vào mỗi R,B,G của điểm ảnh đó. Giá trị của các điểm ảnh đó sẽ được làm tròn lại về số nguyên và được cài đặt để không lớn hơn 255

và nhỏ hơn 0, nếu lớn hơn 255 thì sẽ cài lại mặc định là 255 còn nhỏ hơn 0 thì sẽ mặc định là 0.

3. Hàm FlipDown - FlipLeft (Lật ảnh dọc - ngang)

- Ý tưởng: đảo chiều các cột và các hàng trong ma trận các điểm ảnh.
- Input: đầu vào sẽ là một ảnh(img) .
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó ta sử dụng hai công thức có sẵn trong numpy là $[:,::-1]$ và $[:,::-1, ...]$ để đảo chiều ma trận các điểm ảnh theo chiều dọc và ngang.

4. Hàm Grayscale1 - Grayscale2 (Chuyển đổi ảnh RGB thành ảnh xám)

- Ý tưởng:
 - + Cách 1: với từng điểm ảnh, lấy trung bình giá trị R, G, B rồi gán cho điểm ảnh đó.
 - + Cách 2: với từng điểm ảnh, lấy giá trị R, G, B rồi tính theo công thức có sẵn $(R*0.3 + B*0.59 + G*0.11)$ và gán cho điểm ảnh đó.
- Input: đầu vào sẽ là một ảnh(img) .
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó ta sẽ xét từng điểm ảnh trên từng hàng từng cột lấy ra các giá trị R,B,G. Với 3 giá trị R, G, B của từng điểm ảnh ta tính trung bình cộng hoặc theo công thức có sẵn rồi gán cho từng điểm ảnh đấy. Các giá trị sau khi tính đều sẽ được làm tròn về số nguyên.

5. Hàm Stack2image (Chồng 2 ảnh cùng kích thước: chỉ làm trên ảnh xám)

- Ý tưởng: Thực hiện cộng hai mảng ma trận các điểm ảnh.
- Input: đầu vào sẽ là hai ảnh(img1, img2) có cùng kích thước và là ảnh xám.
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: hai ảnh đưa vào sẽ được chuyển sang 2 ma trận các điểm ảnh sau đó ta thực hiện cộng hai ma trận ấy. Với các điểm ảnh có giá trị lớn hơn 255 thì sẽ cài lại mặc định là 255 còn nhỏ hơn 0 thì sẽ mặc định là 0.

6. Blur (Làm mờ ảnh)

- Ý tưởng: trong ma trận các điểm ảnh, ta phân thành nhiều ma trận vuông bậc 3. Với mỗi ma trận vuông ấy, ta cộng lại tất cả các điểm ảnh lại với nhau rồi nhân cho $1/9$ rồi gán nó cho điểm ảnh tâm của ma trận vuông được chọn ra.
- Input: đầu vào sẽ là một ảnh(img) và một số nguyên(BlurLevel) để có thể điều chỉnh độ làm mờ.

- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh. Ta bắt đầu chạy vòng lặp nhưng sẽ không lấy dòng đầu tiên và cuối cùng của hàng và cột. Với mỗi lần lặp, ta sẽ gán cho tâm điểm ảnh của ma trận vuông được chọn ra bằng giá trị trung bình của 9 điểm ảnh trong ma trận vuông ấy và cứ tiếp tục thế cho tới điểm ảnh cuối.

7. Hàm Circle (Cắt nội dung ảnh theo khung được áp lên (hình tròn))

- Ý tưởng: phương trình đường tròn có tâm và bán kính cho trước:

$(x-a)^2+(y-b)^2 = R^2$ với (a,b) là tâm của ma trận ảnh, R là bán kính. Ta dựa theo công thức để tìm ra các điểm ảnh nằm ngoài đường tròn rồi gán các điểm ảnh đấy bằng 0 thì ta sẽ được một bức ảnh được cắt theo khung tròn.

- Input: đầu vào sẽ là một ảnh(img).
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó tìm bán kính của đường tròn trong ma trận ảnh đó bằng cách chia đôi chiều cao hoặc chiều rộng của ma trận ảnh. Ta duyệt từng điểm ảnh trong ma trận theo phương trình sau: $(i-x)^2+(j-x)^2 > x^2$ (x là bán kính của đường tròn), nếu điểm ảnh $img[i][j]$ sau khi tính có giá trị lớn hơn x^2 thì điểm ảnh ấy nằm ngoài đường tròn, ta sẽ gán điểm ảnh đó bằng 0 còn nằm trong thì ta giữ nguyên giá trị của điểm ảnh đó.

8. Hàm Elip (Cắt nội dung ảnh theo khung được áp lên (hình elip))

- Ý tưởng: phương trình elip có công thức:

$$\frac{(x \cos \alpha + y \sin \alpha)^2}{a^2} + \frac{(x \sin \alpha - y \cos \alpha)^2}{b^2} = 1$$

. Ta dựa theo công thức để tìm ra các điểm ảnh nằm ngoài đường elip rồi gán các điểm ảnh đấy bằng 0 thì ta sẽ được một bức ảnh được cắt theo khung elip. Tiếp tục làm như vậy theo chiều ngược lại thì ta sẽ có hai hình elip chéo nhau.

- Input: đầu vào sẽ là một ảnh(img) .
- Output: đầu ra sẽ là ma trận các điểm ảnh đã xử lý(img).
- Mô tả: ảnh đưa vào sẽ được chuyển sang 1 ma trận các điểm ảnh sau đó tìm khoảng cách từ tâm đến các cạnh trong ma trận ảnh đó bằng cách chia đôi chiều cao hoặc chiều rộng của ma trận ảnh. Ta duyệt từng điểm ảnh trong ma trận trên 2 đường elip bằng cách sau:

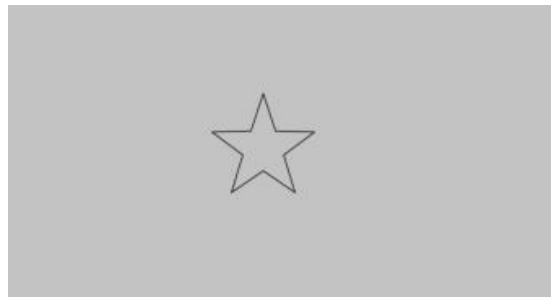
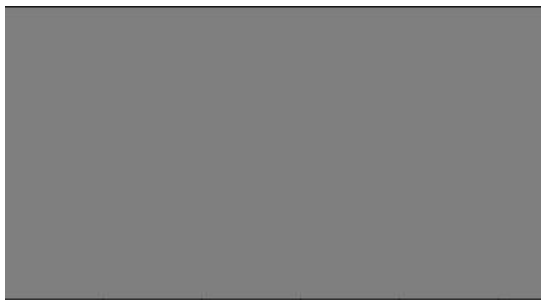
```
if ((i - x)*cos + (j - x) * SIN)**2 / (x*x) + ((i - x)*SIN - (j - x)*COS)**2 / ((x/2)**2) > 1:
    if ((i - x)*cos + (j - x) * SIN)**2 / ((x/2)**2) + ((i-x)*SIN - (j-x)*COS)**2 / (x*x) > 1:
        img[i][j]=0
```

Với x là khoảng cách, cos(alpha) và sin(alpha) bằng cos(45) nếu điểm ảnh $img[i][j]$ sau khi tính có giá trị lớn hơn 1 trên cả hai đường elip thì điểm ảnh ấy nằm ngoài hai đường elip, ta sẽ gán điểm ảnh đó bằng 0 còn nằm

trong thì ta giữ nguyên giá trị của điểm ảnh đó. Ảnh chỉ đúng khi ma trận là ma trận vuông.

III. Hình ảnh kết quả với từng chức năng:

*** Ảnh ban đầu**



1. Brightness

- Bright = 150



2. Contrast

- Contrast = 150

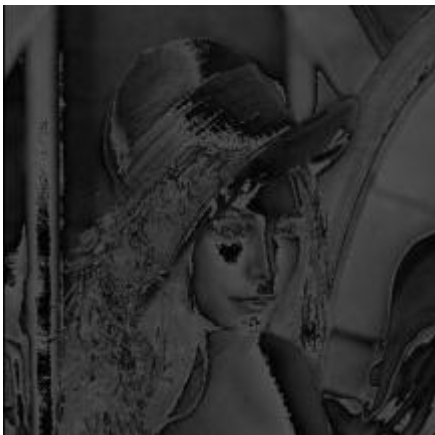


3. Grayscale

*Grayscale1



*Grayscale2



4. FlipDown - FlipLeft

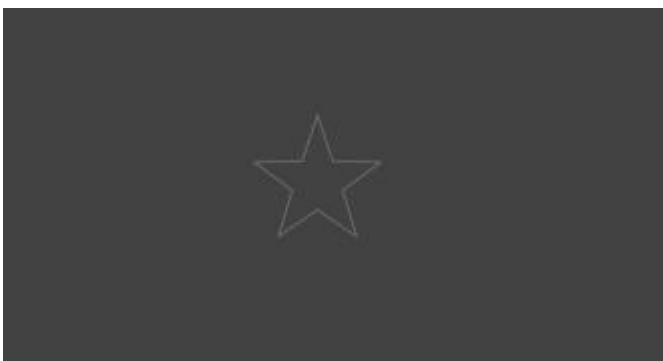
*FlipDown



*FlipLeft

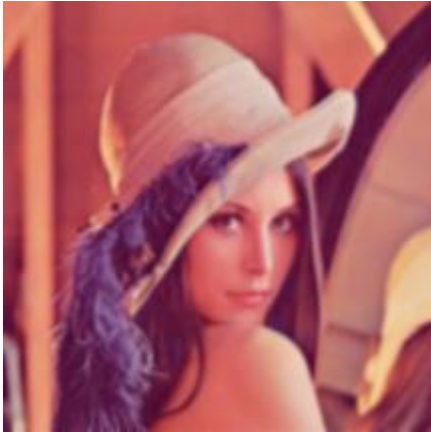


5. Stack2image (Chồng hai ảnh xám)



6. Blur

- BlurLevel = 5











7. Circle



8. Elip



9. Thực hiện tất cả chức năng

	Blur.png	Type: PNG File Dimensions: 512 x 512	Size: 380 KB
	Brightness.png	Type: PNG File Dimensions: 512 x 512	Size: 301 KB
	Circle.png	Type: PNG File Dimensions: 512 x 512	Size: 447 KB
	Contrast.png	Type: PNG File Dimensions: 512 x 512	Size: 350 KB
	Elip.png	Type: PNG File Dimensions: 512 x 512	Size: 530 KB
	Flip.png	Type: PNG File Dimensions: 512 x 512	Size: 547 KB
	Grayscale.png	Type: PNG File Dimensions: 512 x 512	Size: 258 KB
	Stackimage.png	Type: PNG File Dimensions: 2742 x 1470	Size: 23.0 KB

IV. Tài liệu tham khảo

* Tài liệu tham khảo của các chức năng: 1, 2, 3, 4, 5, 6

<https://koodibar.com/posts/xu-ly-hinh-anh-voi-python#11-code-th%E1%BB%A7-c%C3%B4ng>

<https://stackoverflow.com/questions/51285593/converting-an-image-to-grayscale-using-numpy>

<https://www.dfstudios.co.uk/articles/programming/image-programming-algorithms/image-processing-algorithms-part-5-contrast-adjustment/>

<https://www.w3resource.com/numpy/manipulation/flipr.php>

https://en.wikipedia.org/wiki/Box_blur?fbclid=IwAR1K_oO4Awa-CIpa8RAM3xmew2lueU1ZFbC-IeECMCf6-OpBxw9iDuNBxdY

* Tài liệu tham khảo của các chức năng: 7, 8

<https://loigiaihay.com/ly-thuyet-phuong-trinh-duong-tron-c45a5233.html>

https://www.maa.org/external_archive/joma/Volume8/Kalman/General.htm?fbclid=IwAR1XMiSPZyJBxZHLZw2TfpwcTfmvmpwLUNJiO46PqCFyHgVndZBgKf-cM