

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

ĐỒ ÁN 2 – IMAGE PROCESSING



Bộ môn : Toán ứng dụng và thống kê cho Công nghệ thông tin

Giảng viên lý thuyết : Vũ Quốc Hoàng

Giảng viên thực hành : Nguyễn Văn Quang Huy

Trần Thị Thảo Nhi

Phan Thị Phương Uyên

Sinh viên thực hiện

Lớp : 20CLC05

MSSV : 20127206

Tên : Vũ Đình Duy Khánh

MỤC LỤC

I.	Thông tin cá nhân	2
II.	Các chức năng đã hoàn thành.....	2
III.	Mô tả các hàm chức năng.....	2
1.	Thay đổi độ sáng cho ảnh	2
2.	Thay đổi độ tương phản	2
3.	Lật ảnh ngang – dọc	3
4.	Chuyển đổi ảnh RGB thành ảnh xám	3
5.	Chồng 2 ảnh cùng kích thước	3
6.	Làm mờ ảnh	4
7.	Khung ảnh	4
IV.	Hình ảnh kết quả.....	5
	Ảnh gốc :	5
1.	Thay đổi độ sáng cho ảnh :	5
2.	Thay đổi độ tương phản :.....	5
3.	Lật ảnh.....	6
4.	Chuyển ảnh RGB thành ảnh xám.....	6
5.	Chồng 2 ảnh cùng kích thước	6
6.	Làm mờ	7
7.	Cắt ảnh theo khung	7
V.	Tài liệu tham khảo	7

I. Thông tin cá nhân

Lớp : 20CLC05

MSSV : 20127206

Tên : Vũ Đình Duy Khánh

II. Các chức năng đã hoàn thành

STT	Chức năng	Mức độ 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	100%
6	Làm mờ ảnh	100%
7	Cắt ảnh theo khung	100%

III. Mô tả các hàm chức năng

1. Thay đổi độ sáng cho ảnh

- Ý tưởng : Tăng/Giảm giá trị tất cả các điểm ảnh để tăng/giảm độ sáng cho ảnh
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
 - alpha (float) : giá trị muốn tăng/giảm độ sáng
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array
- Mô tả : Tăng/Giảm độ sáng bằng cách cộng/trừ tất cả điểm ảnh với hệ số alpha và giới hạn trong khoảng [0,255]

2. Thay đổi độ tương phản

- Ý tưởng : Tăng/Giảm độ tương phản của ảnh bằng cách tăng/giảm khoảng cách giữa các giá trị màu sáng, tối
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
 - alpha (float) : giá trị muốn tăng/giảm độ sáng
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array

- Mô tả : Nhân tất cả các điểm ảnh với công thức $R' = F(R - 128) + 128^1$, với R là giá trị điểm ảnh

3. Lật ảnh ngang – dọc

- Ý tưởng : Đổi vị trí của các hàng/cột đối xứng nhau qua các trục dọc/ngang nằm chính giữa tấm ảnh
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
 - type (string) : chiều ảnh muốn lật vertical/horizontal
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array
- Mô tả :
 - type == 'vertical' : sử dụng hàm np.fliplr để lật ảnh ngang theo trục dọc giữa tấm ảnh
 - type == 'horizontal' : sử dụng hàm np.flipud để lật ảnh dọc theo trục ngang giữa tấm ảnh

4. Chuyển đổi ảnh RGB thành ảnh xám

- Ý tưởng : tăng/giảm số lượng màu Red, Green, Blue trong các điểm ảnh để nhận được một hình ảnh có thang màu xám theo 2 phương pháp : phương pháp trung bình và phương pháp nhân trọng số²
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
 - type (string) : phương pháp chuyển đổi ảnh RGB thành ảnh xám average/weight
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array
- Mô tả :
 - Phương pháp trung bình : tất cả điểm ảnh lấy giá trị trung bình của 3 màu : $(R+G+B)/3$
 - Phương pháp nhân trọng số : mỗi điểm ảnh có mới giá trị bằng cách giảm Red, tăng Green, thêm Blue : $(0.3*R) + (0.59*G) + (0.11*B)$

5. Chồng 2 ảnh cùng kích thước

- Ý tưởng : chuyển ảnh thành ảnh xám, sau đó cộng 2 ảnh lại với nhau (mỗi ảnh lấy 1 nửa giá trị trên dải màu)
- Input :
 - image1 (np.array) : ảnh đầu vào 1 dưới dạng numpy array
 - image2 (np.array) : ảnh đầu vào 2 dưới dạng numpy array
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array

- Mô tả :
 - Dùng hàm chuyển ảnh RGB thành ảnh xám đã làm ở mục 4 để chuyển 2 ảnh thành ảnh xám.
 - Cộng một nửa dải màu ảnh 1 với một nửa dải màu ảnh 2

6. Làm mờ ảnh

- Ý tưởng : Mỗi điểm ảnh mới có giá trị bằng trung bình của chính nó là 8 điểm ảnh xung quanh
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array
- Mô tả :
 - Xử dụng thuật toán Box Blur³ (trung bình tổng 8 điểm ảnh xung quanh và chính nó) tạo thành điểm ảnh mới

7. Khung ảnh

- Ý tưởng :
 - Hình tròn : những điểm ảnh nằm bên ngoài hình tròn có tâm là tâm của ảnh, đường kính là chiều dài/chiều rộng (giá trị bé hơn) được chuyển thành màu đen
- Input :
 - image (np.array) : ảnh đầu vào dưới dạng numpy array
 - type (string) : khung muốn cắt circle/DoubleEllipses
- Output :
 - newImage (np.array) : ảnh kết quả đầu ra dưới dạng numpy array
- Mô tả :
 - Hình tròn : chạy từng điểm ảnh, kiểm tra nếu nằm ngoài hình tròn set giá trị màu của nó thành 0
 - Hình elip : chạy từng điểm ảnh, kiểm tra nếu nằm ngoài cả 2 phương trình elip⁴ thì set giá trị màu của nó thành 0

IV. Hình ảnh kết quả

Ảnh gốc :



1. Thay đổi độ sáng cho ảnh :



-100 brightness



+100 brightness

2. Thay đổi độ tương phản :



-100 contrast



+100 contrast

3. Lật ảnh



Lật ngang

Lật dọc

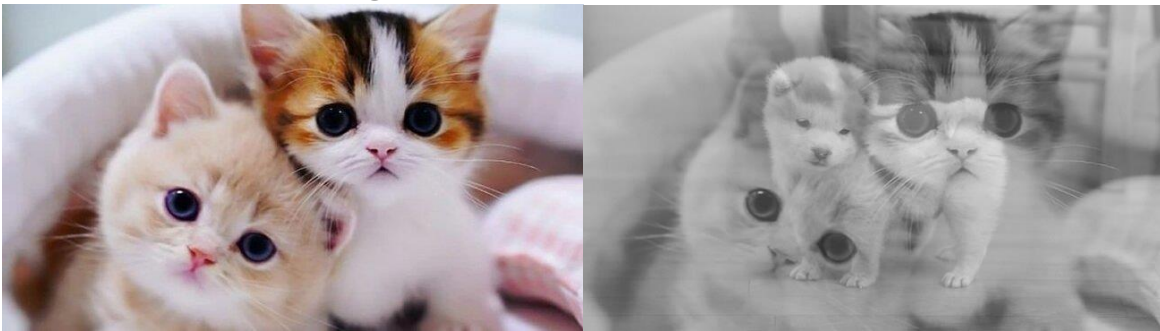
4. Chuyển ảnh RGB thành ảnh xám



type = 'average'

type = 'weight'

5. Chồng 2 ảnh cùng kích thước



Ảnh 2

Kết quả

6. Làm mờ



Kết quả

7. Cắt ảnh theo khung



Khung tròn



Khung elip

V. Tài liệu tham khảo

¹ [Image Processing Algorithms Part 5: Contrast Adjustment | Dreamland Fantasy Studios \(dfstudios.co.uk\)](https://www.dfstudios.co.uk/blog/image-processing-algorithms-part-5-contrast-adjustment/)

² [Grayscale to RGB Conversion \(tutorialspoint.com\)](https://www.tutorialspoint.com/opencv/opencv_image_processing_10.htm)

³ [Kernel \(image processing\) - Wikipedia](https://en.wikipedia.org/wiki/Kernel_(image_processing))

⁴ [The Most Marvelous Theorem in Mathematics \(maa.org\)](https://maa.org/mathematics/the-most-marvelous-theorem-in-mathematics)