

**UNIVERSITY OF SCIENCE
FALCUTY OF INFORMATION TECHNOLOGY**



SUBJECT: Applied Mathematics and Statistics

PROJECT 2: IMAGE PROCESSING

Name: Nguyễn Quốc Huy

Class: 20CLC02

Student-ID: 20127188

Lecturers: **VŨ QUỐC HOÀNG**
NGUYỄN VĂN QUANG HUY
TRẦN THỊ THẢO NHI
PHAN THỊ PHƯƠNG UYÊN

Mục lục

I. Yêu cầu đề án	3
II. Ý tưởng thực hiện	3
III. Mô tả các hàm	4
1) Khai báo thư viện và hàm cần thiết	4
2) Hàm brightness – Thay đổi độ sáng hình ảnh	4
3) Hàm contrast – Thay đổi độ tương phản hình ảnh	4
4) Hàm flip – Lật ảnh ngang dọc	4
5) Hàm gray_scale – Chuyển đổi ảnh RGB về ảnh xám	5
6) Hàm add_gray_picture – Cộng 2 ảnh xám cùng kích thước	5
7) Hàm box_blur – Làm mờ hình ảnh	5
8) Hàm main	5
IV. KẾT QUẢ	6
V. BẢNG CÔNG VIỆC	10
VI. NGUỒN THAM KHẢO	10

I. Yêu cầu đề án

Trong đề án này, yêu cầu thực hiện các chức năng xử lý ảnh cơ bản sau:

- Thay đổi độ sáng cho ảnh
- Thay đổi độ tương phản
- Lật ảnh (ngang - dọc)
- Chuyển đổi ảnh RGB thành ảnh xám
- Chồng 2 ảnh cùng kích thước (thực hiện trên ảnh xám)
- Làm mờ ảnh
- Viết hàm main xử lý
- Các thư viện được sử dụng PIL (open(), save() từ Image) để đọc và ghi; Matplotlib (imshow() từ pyplot) để hiển thị ảnh.
- Được phép sử dụng thư viện NumPy tùy ý.
- Lưu ý: Để được **điểm tối đa** cho từng chức năng, thời gian thực thi của mỗi chức năng phải nằm trong khoảng thời gian chấp nhận được. Ví dụ với chức năng làm mờ (phức tạp nhất) có thời gian thực thi trên ảnh với kích thước 512x512 là dưới 20 giây.

II. Ý tưởng thực hiện

• Thay đổi độ sáng cho ảnh

Để thay đổi độ sáng cho ảnh thì điều chúng ta cần làm là tăng giá trị của điểm ảnh theo thông số mà người dùng nhập vào, ngoài ra cũng phải giới hạn giá trị điểm ảnh nằm trong khoảng từ [0,255]

• Thay đổi độ tương phản

Tương tự với thay đổi độ sáng thì độ tương phản là khác biệt giữa màu sáng và màu tối nên chúng ta chỉ cần tăng giá trị theo công thức để làm tăng sự khác biệt giữa 2 màu sáng tối

• Lật ảnh (ngang dọc)

Với yêu cầu này thì chúng ta chỉ cần chuyển hóa vị trí của ma trận thì chúng ta sẽ có được một bức hình được lật

• Chuyển đổi ảnh RGB về ảnh xám

Ảnh xám khác với ảnh RGB là 1 pixel chỉ có một chỉ điểm màu duy nhất thay vì có 3 phần tử R G B như thông thường nên việc chúng ta cần làm là đưa nó về một thông số màu duy nhất bởi công thức cô đã giới thiệu trên lớp

• Chồng 2 ảnh xám cùng kích thước

Yêu cầu này chúng ta sẽ lấy trung bình cộng của 2 ảnh xám

• Làm mờ ảnh

Ở đây em dùng phương pháp box blur, trong đó mỗi pixel hình ảnh là kết quả có giá trị bằng giá trị trung bình của các pixel lân cận của nó trong hình ảnh đầu vào

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

III. Mô tả các hàm

1) Khai báo thư viện và hàm cần thiết

- Ở đây em dùng 3 thư viện chính mà cô đã yêu cầu từ trước là numpy, PIL và matplotlib với numpy dùng để xử lý mảng đa chiều trong hình ảnh, giới hạn phạm vi điểm ảnh, ép kiểu dữ liệu, thư viện PIL em dùng để mở ảnh là chủ yếu và thư viện matplotlib em dùng để hiển thị và lưu hình ảnh
- Em khai báo thêm một hàm cần thiết đó là hàm truncate, hàm này có chức năng là giới hạn điểm ảnh chỉ nằm trong khoảng từ [0, 255] và em sẽ dùng hàm np.where^[1] để thực hiện chức năng giới hạn, nếu giá trị của pixel vượt quá 255 thì sẽ gán lại bằng 255 và nhỏ hơn 0 sẽ gán lại bằng 0 còn lại nếu nằm trong khoảng 0 -> 255 thì sẽ giữ nguyên

2) Hàm brightness – Thay đổi độ sáng hình ảnh

- Như ý tưởng em nêu bên trên thì hàm này em truyền vào 2 thông số đó là ảnh và mức độ sáng tối mà người dùng muốn chỉnh ảnh. Em thực hiện cộng giá trị các điểm ảnh với level. Nếu người dùng nhập giá trị dương thì hàm sẽ sáng và ngược lại với giá trị âm, đồng thời sử dụng hàm truncate khai báo bên trên để giới hạn lại điểm ảnh trong phạm vi [0,255]

3) Hàm contrast – Thay đổi độ tương phản hình ảnh

- Để thực hiện hàm này đầu tiên em tính toán hệ số tương phản $F^{[2]}$ với công thức $F = 259 * (255 + level) / 255 * (259 - level)$ với level là giá trị tương phản mà người dùng truyền vào, giá trị F em sẽ làm tròn lên số nguyên với hàm round() vì giá trị của điểm ảnh phải là số nguyên
- Sau đó sử dụng hàm truncate để giới hạn lại các điểm ảnh và trả về ma trận

4) Hàm flip – Lật ảnh ngang dọc

- Hàm này em sử dụng hàm có sẵn trong thư viện PIL là Image.FLIP_LEFT_RIGHT^[3] để thay đổi theo chiều dọc và

Image.FLIP_TOP_BOTTOM để thay đổi theo chiều ngang và Image.transpose() để chuyển vị ma trận và sau đó trả về ma trận

5) Hàm gray_scale – Chuyển đổi ảnh RGB về ảnh xám

- Do các pixel của ảnh xám khác với ảnh RGB thông thường nên chúng ta duyệt từng giá trị pixel của ảnh RGB sau đó chuyển về pixel của ảnh xám thông qua công thức^[4] $0.2989 * r + 0.5870 * g + 0.1140 * b$ (3 thông số này tương tự với 0.3R, 0.59G, 0.11B mà cô đã giới thiệu trong đề án)

6) Hàm add_gray_picture – Cộng 2 ảnh xám cùng kích thước

- Trong hàm này thì người dùng có thể chọn 2 bức ảnh tùy ý, có thể là ảnh màu hoặc ảnh xám nhưng phải cùng kích thước, em sẽ sử dụng lại hàm gray_scale để làm xám 2 ảnh được truyền vào sau đó dùng cộng một $\frac{1}{2}$ ma trận để tránh tràn số, em dùng hàm cộng ma trận np.add() để trả về ma trận mới

7) Hàm box_blur – Làm mờ hình ảnh

- Như ý tưởng em đã nêu bên trên^[5], trong thuật toán box_blur mỗi pixel x có một giá trị bằng với giá trị trung bình từ ma trận vuông 3×3 với tâm tại x. Nên em sẽ duyệt qua từng pixels ảnh để tính được tổng các hệ màu từ đó chia cho 9 để có được giá trị trung bình, sau đó dùng hàm truncate để giới hạn lại các điểm ảnh, gán lại các hệ màu mới và trả về ma trận

8) Hàm main

- Hàm main của em chỉ đơn giản là viết cấu trúc điều kiện, cho người dùng nhập vào lựa chọn thực hiện tính năng với
 - 0. Thực hiện toán bo
 - 1. Thay đổi độ sáng
 - 2. Thay đổi tương phản
 - 3. Xoay ảnh dọc, ngang
 - 4. Thay đổi ảnh từ RGB thành ảnh xám
 - 5. Gộp 2 ảnh xám có cùng kích thước
 - 6. Làm mờ ảnh
- Tùy vào người dùng nhập thì sẽ rẽ nhánh vào các hàm riêng, trong mỗi hàm em dùng Image.open() để mở ảnh, sử dụng hàm find để tìm ra tên file mà người dùng nhập vào (sử dụng cho yêu cầu lưu ảnh), lấy ra từng pixels sau đó người dùng sẽ nhập vào con số tương ứng thỏa mãn yêu cầu
 - Với yêu cầu thay đổi độ sáng và tương phản thì người dùng sẽ nhập vào độ sáng / tương phản mà người dùng thích (Lưu ý độ tương phản chỉ nhập từ 0 -> 255)

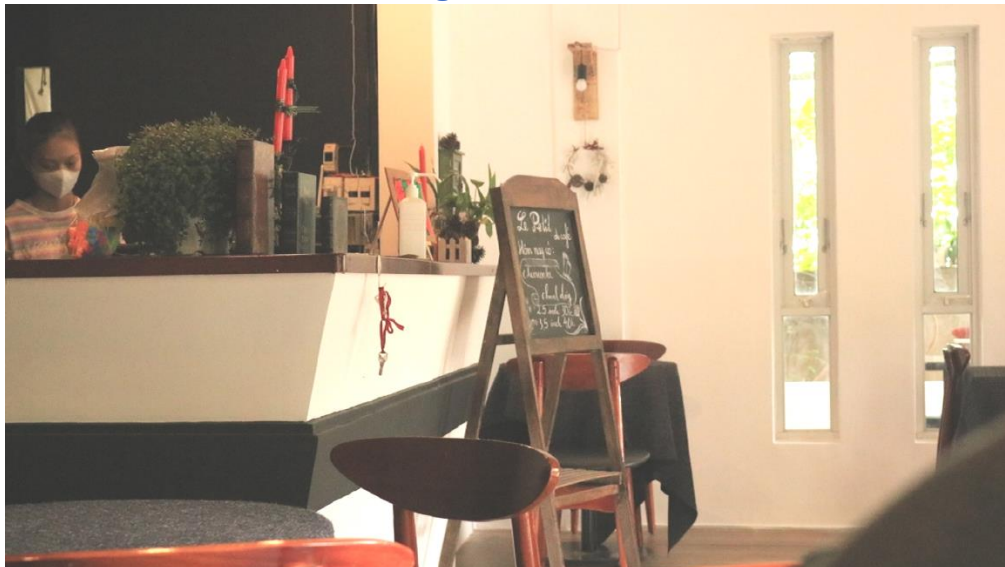
- Với yêu cầu xoay ảnh dọc ngang người dùng sẽ được chọn 2 lựa chọn một là xoay dọc còn 2 là xoay ngang
- Sau khi đã nhập hoàn tất thì hàm sẽ chạy và trả về kết quả bằng hàm plt.imshow() và lưu ảnh bằng hàm plt.imsave()

IV. KẾT QUẢ

Ảnh gốc



Brightness = 50



Contrast = 50



Flip vertical / horizontal





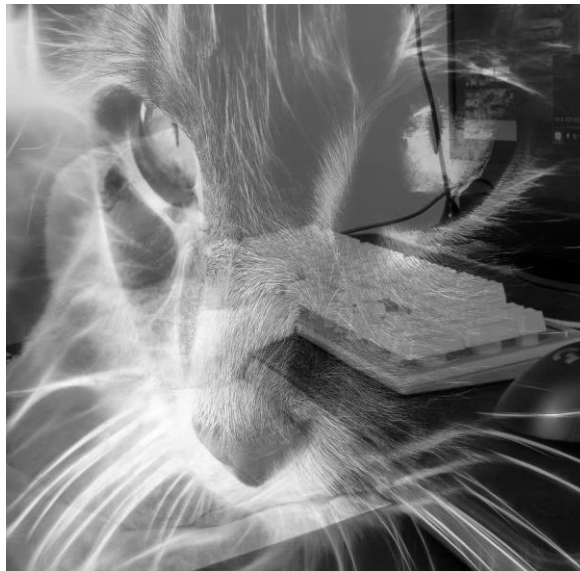
Gray



Chồng 2 ảnh xám



=



Blur



V. BẢNG CÔNG VIỆC

Công việc	Hoàn thành
Thay đổi độ sáng hình ảnh	100%
Thay đổi độ tương phản hình ảnh	100%
Lật ảnh ngang dọc	100%
Chuyển đổi ảnh RGB về ảnh xám	100%
Chồng 2 ảnh xám cùng kích thước	100%
Làm mờ ảnh	100%
Main	100%

VI. NGUỒN THAM KHẢO

- [1] <https://numpy.org/doc/stable/reference/generated/numpy.where.html>
- [2] <https://stackoverflow.com/questions/42045362/change-contrast-of-image-in-pil>
- [3] <https://pythonexamples.org/python-pillow-flip-image-vertical-horizontal/#:~:text=To%20flip%20an%20Image%20vertically,on%20the%20PIL%20Image%20object.>
- [4] <https://stackoverflow.com/questions/12201577/how-can-i-convert-an-rgb-image-into-grayscale-in-python>
- [5] <https://stackoverflow.com/questions/65141759/box-blur-is-not-any-faster-than-gaussian-blur>