Nguyễn Quang Trường
19127608

Báo cáo đồ án

Các chức năng đã hoàn thành

- Thay đổi độ sáng.
- Thay đổi độ tương phản.
- Chuyển đổi ảnh RGB thành ảnh xám.
- Lật ảnh (ngang – dọc)
- Chồng 2 ảnh có cùng kích thước.
- Làm mờ ảnh.

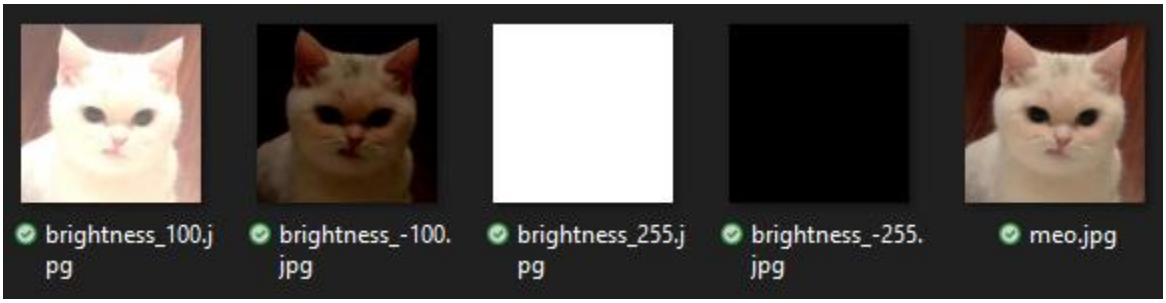
Ý tưởng thực hiện và mô tả

1. **Thay đổi độ sáng**
   Tăng hoặc giảm giá trị của mảng hình ảnh đã cho dựa trên số liệu người dùng nhập vào, nếu một điểm ảnh giảm xuống dưới 0 thì giá trị đó vẫn là 0, nếu quá 255 thì vẫn là 255.

```
D:\OneDrive - VNU-HCMUS\Toan_Ung_Dung\Pr_02>python 19127608.py
Adjusting brightness on meo.jpg
Enter number to adjust brightness: 2000
Output result to brightness_255.jpg
```

Ở đây, ta thay đổi với giá trị độ sáng tăng thêm 2000, vậy tất cả điểm ảnh sẽ là 255 vì độ sáng sẽ trong khoảng [0, 255]

Tương tự, dưới đây là một số ví dụ, số liệu thay đổi đã được biểu thị trong tên file ảnh theo cấu trúc brightness_<giá trị độ sáng>.jpg



2. **Thay đổi độ tương phản**

Để thay đổi độ tương phản, ta cần tính hệ số hiệu chỉnh theo công thức

$$F = \frac{259(C + 255)}{255(259 - C)}$$

Nếu F = 1, ảnh sau khi thay đổi độ tương phản sẽ giống ảnh cũ đã cho.

Sau đó, ta có thể thay đổi độ tương phản từng màu theo công thức dưới đây:

Ví dụ như màu đỏ:

$$R' = F(R - 128) + 128$$

Tuy nhiên, dựa vào thư viện numpy, ta có thể thay đổi cùng lúc cả ba màu R, G, B dựa vào hàm np.clip()

Cùng lúc, ta giới hạn độ tương phản sau khi thay đổi trong khoảng [0, 255]

Dưới đây là hình ảnh minh họa, độ tương phản đã được biểu thị rõ trong tên file ảnh với cấu trúc contrast_<giá trị độ tương phản>.jpg

3. **Chuyển đổi ảnh RGB thành ảnh xám**

Chuyển đổi ảnh RGB sang ảnh xám sử dụng format luma của CCIR 601 (Định dạng hầu như tiêu chuẩn của ảnh, video kĩ thuật số)

Ta áp dụng công thức dưới đây cho mọi điểm ảnh để thay đổi màu cả ba hệ RGB khiến tổ hợp 3 màu được ghi nhận là màu xám (đối với con người).

$$I = 0.299R + 0.587G + 0.114B$$

Hình ảnh minh họa:

4. **Lật ảnh (ngang – dọc)**

Ta chỉ cần đảo ngược từng hàng hoặc cột điểm ảnh trong ma trận điểm ảnh đã cho.
Với python, ta có thể sử dụng cơ chế cắt lát (slice) đối với ma trận để thay đổi ma trận một cách nhanh gọn chỉ với một dòng.

```
left_right = image[:, ::-1, :] #reverse rows
up_down = image[::-1] #reverse columns
```

Hình ảnh minh họa:

5. **Chồng 2 ảnh có cùng kích thước**

Việc chồng hai ảnh có ràng buộc đó là:

- Chỉ thực hiện trên ảnh xám.
- Hai ảnh có cùng kích thước.

Vậy nên, hai file đính kèm để thực hiện chồng ảnh đã được chuyển thành ảnh xám trước đó, hàm chồng ảnh không có chức năng chuyển ảnh màu về ảnh xám mà chỉ và chỉ chồng 2 ảnh được cho.
Vì hai ảnh được cung cấp sẵn không có cùng độ phân giải nên em sẽ chỉnh cả hai về 200x200, tức 200 điểm ảnh ở cột và hàng.
Sau đó, chồng ảnh theo phương pháp alpha với giá trị alpha có thể thay đổi trong khoảng [0, 1]
Ví dụ:

Ta sẽ chồng hai ảnh này lại làm một.

Kết quả với **alpha = 0.01**

Kết quả với **alpha = 0.5**

Kết quả với **alpha = 0.7**

Kết quả với **alpha = 0.99**

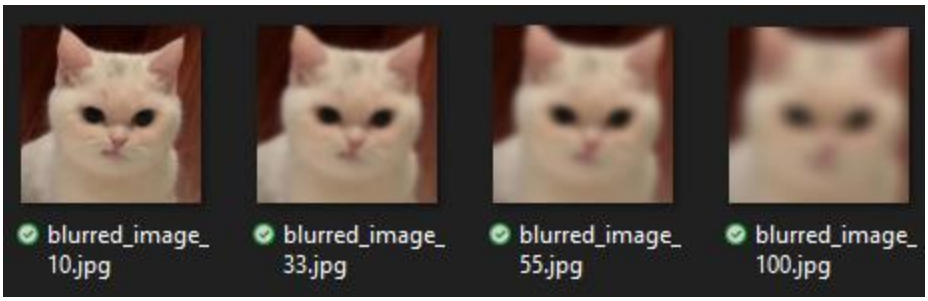
6. **Làm mờ ảnh**

   Ta sử dụng phương pháp làm mờ Gaussian (cũng có thể gọi là làm mịn Gaussian) bằng công thức Gaussian, còn gọi là phân phối chuẩn trong thống kê.
   Đầu tiên, ta tìm kernel của bộ lọc Gaussian với sigma có giá trị = (<giá trị làm mờ> - 1) / 6
   Sau đó, ta tính tích chập trên ảnh gồm 3 kênh màu RGB bằng cách tính tích chập trên từng ma trận 2 chiều sau đó gộp lại về ma trận 3 kênh màu RGB.
   Công thức tính tích chập giữa hàm ảnh f(x, y) và bộ lọc (filter) k(x, y) cho ảnh có kích thước mxn:

$$k(x,y)\bigstar f(x,y) = \sum_{u=-m/2}^{m/2}\sum_{v=-n/2}^{n/2} k(u,v)f(x-u,y-v)$$

   Hình ảnh minh họa với giá trị làm mờ được biểu thị rõ thông qua tên file ảnh với cấu trúc blurred_image_<giá trị làm mờ>.jpg với giá trị làm mờ > 0

Nguyễn Quang Trường
19127608



Tài liệu tham khảo

Image processing algorithms

Image processing - blurring

Gaussian blur

Phép tích chập trong xử lý ảnh

Understanding slice notation

Alpha blending and masking of images with Python, OpenCV, NumPy