

Toán Ứng Dụng Thống Kê

Báo Cáo Project 2

Chủ đề: Image Processing

Lớp: 21CLC05

Giảng viên thực hành:

- Nguyễn Văn Quang Huy
- Phan Thị Phương Uyên

Thông tin sinh viên:

- 20127030 - Nguyễn Mạnh Hùng

Mục Lục

1. Giới thiệu đề án	2
2. Quy trình thực thi của chương trình	2
3. Ý tưởng thực hiện	3
4. Mô tả chi tiết các hàm	5
5. Khảo sát kết quả thực thi	7
6. Nhận xét chung và kết luận	10
7. Nguồn tài liệu tham khảo	10

I. Giới thiệu đề án

Ở đề án kỳ này, em đã xây dựng hệ thống nén xử lý ảnh. Với ảnh dữ liệu đầu vào, ta lần lượt tiến hành đọc file ảnh và tiến hành thực hiện các chức năng cơ bản theo của người dùng bao gồm:

- Chỉnh sửa độ sáng.
- Chỉnh sửa độ tương phản
- Lật ảnh ngang/dọc
- Chuyển ảnh thành ảnh xám/sepia
- Làm mờ/sắc nét ảnh
- Cắt ảnh theo kích thước trung tâm
- Cắt ảnh theo khung tròn

Bài tập được thực hiện bằng ngôn ngữ lập trình Python phiên bản 3.11.4 định dạng ở file jupyter notebook (.ipynb) và các thư viện được dùng cho đề án này là Numpy, Pillow và Matplotlib

II. Quy trình thực thi của chương trình

Do thực hiện trên file jupyter notebook để thuận tiện cho việc so sánh, đối chiếu các kết quả hình ảnh, thời gian chạy và chú thích các ô code nên chương trình của em sẽ được thực hiện theo quy trình sau:

- Bước 1: cài đặt các thư viện cần thiết và nạp các hàm cần thiết cho chương trình.
- Bước 2: cho người dùng nhập tên file ảnh (*Lưu ý: cần phải nhập đúng tên/đường dẫn + định dạng ảnh thì chương trình mới thực thi đúng*).
- Bước 3: hiển thị thành menu và cho phép người dùng nhập vào các chức năng tương ứng ở đây. Lưu ý: nhập số 0 để thực hiện hết toàn bộ chức năng của hệ thống.
- Bước 4: khi thực hiện xong thì xuất ra kết quả ảnh ở màn hình + các thông số kích cỡ ảnh, đồng thời lưu ảnh kết quả ở định dạng <tên file ảnh gốc>_<tên chức năng thực hiện>.jpg

III. Ý tưởng thực hiện

Ở phần này ta lần lượt đi tìm hiểu các nguyên lý cơ bản của xử lý ảnh.

1. Chỉnh sửa độ sáng.

Bản chất bài toán này là việc đi cộng một ma trận ảnh cho một số thực, việc tăng hay giảm các giá trị màu của điểm ảnh sẽ ảnh hưởng tới việc độ sáng hay tối của bức ảnh đấy.

Công thức:

$$\text{new_img} = \text{img} + a$$

Với *new_img* là ma trận ảnh mới sau khi chỉnh sửa, *img* là ma trận ảnh gốc và *a* là giá trị số nguyên.

2. Chỉnh sửa độ tương phản.

Tương tự với độ sáng, bài toán này cơ bản là phép nhân giữa số thực và ma trận ảnh.

Công thức:

$$\text{new_img} = \text{img} * a$$

Với *new_img* là ma trận ảnh mới sau khi chỉnh sửa độ tương phản, *img* là ma trận ảnh gốc và *a* là giá trị số thực.

3. Lật ảnh dọc – ngang.

Tác vụ này giống như lấy đối xứng của một bức ảnh thông qua trục đối xứng ngang hay dọc. Việc lật ảnh theo trục đối xứng dọc được thực hiện bằng cách, với mỗi cặp cột đối xứng lẫn nhau thông qua trục dọc trung tâm trong ảnh, ta đổi vị trí của các cặp tương ứng với nhau. Tương tự với lật ảnh ngang, với mỗi cặp dòng đối xứng lẫn nhau qua trục hoành trung tâm của ảnh, hoán đổi vị trí của các cặp tương ứng với nhau.

4. Chuyển đổi ảnh màu thành ảnh xám/sepia

Chuyển đổi hình ảnh màu thành hình ảnh có thang độ màu xám bằng cách lấy giá trị trung bình của các giá trị RGB: giảm các điểm ảnh có chứa điểm màu đỏ, điểm màu xanh lá và đặt điểm ảnh có màu xanh dương vào giữa 2 điểm ảnh này, và tăng các điểm ảnh có chứa, được tính theo trọng số ma trận. Từ đó, ta có được công thức như sau:

$$\text{gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

Cũng tương tự như chuyển đổi ảnh xám, chuyển đổi ảnh sepia cũng lấy tổng các kênh màu được tính theo trọng số nhưng ta phải làm đến 3 lần với mỗi lần là 3 bộ trọng số khác nhau để tạo ra kết quả tương ứng cho 3 kênh màu R, G, B mới.

Thuật toán

$\text{newR} = 0.393\text{R} + 0.769\text{G} + 0.189\text{B}$ (lấy trị nguyên)

$\text{newG} = 0.349\text{R} + 0.686\text{G} + 0.168\text{B}$ (lấy trị nguyên)

$\text{newB} = 0.272\text{R} + 0.534\text{G} + 0.131\text{B}$ (lấy trị nguyên)

Set các giá trị cho từng pixels ứng với từng kênh màu kết quả vừa tính được ở trên qua điều kiện:

If $\text{newR} > 255$ then $\text{newR} = 255$ else $\text{newR} = \text{tr}$

If $\text{newG} > 255$ then $\text{newG} = 255$ else $\text{newG} = \text{tg}$

If $\text{newB} > 255$ then $\text{newB} = 255$ else $\text{newB} = \text{tb}$

5. Làm mờ/sắc nét ảnh

Thuật toán làm mờ ảnh thực hiện thông qua việc tính phép toán tích chập giữa ảnh và mặt nạ ảnh (kernel), trong đó mặt nạ ảnh là ma trận Gaussian có kích thước $k \times k$. Phép toán làm mờ ảnh được thể hiện qua công thức sau:

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

Với k là mặt nạ ảnh và f là hàm ảnh phép $k \star f$ là phép tính tích chập giữa ảnh và mặt nạ ảnh.

Thuật toán làm sắc nét ảnh thực hiện bằng cách, đầu tiên ta tính đại lượng chênh lệch ảnh hiệu hai ma trận giữa ảnh gốc với ảnh làm mờ, sau đó ta sẽ thực hiện phép cộng 2 ma trận giữa ảnh gốc và ma trận đại lượng đó. Khi đó, ảnh sẽ được làm sắc nét hơn và rõ hơn. Vì

khi này các pixel mang giá trị nhiều sẽ được tăng cường và các điểm ảnh sẽ được nâng độ đậm của màu lên tương ứng theo phân bố màu ở trong hình.

6. Cắt xén ảnh ở trung tâm

Tính toán các chiều dài và chiều rộng của bức ảnh cần cắt xén theo chiều dài và chiều rộng của ảnh, lấy điểm ảnh trung tâm của ảnh gốc cũng là điểm ảnh trung tâm của ảnh kết quả cắt xén.

7. Cắt ảnh theo khung hình tròn

Tính width, height của ma trận ảnh. Tính bán kính của ma trận ảnh để thực hiện việc tính khung tròn ngoài cần cắt cho ảnh bằng cách lấy một cạnh của ma trận vừa tính được chia đôi. Sau cùng lấy điểm ảnh trung tâm làm gốc, thiết lập phương trình đường tròn quanh điểm gốc. Xét từng điểm trong bức ảnh, nếu thuộc hay nằm trong phương trình đường tròn thì giữ, còn lại thì set giá trị = 0.

8. Thiết lập hàm main

Ta sẽ cho người dùng nhập tên file ảnh vào, sau đó nhập các sự lựa chọn + các thông số (nếu có) để thực hiện các tác vụ xử lý tương ứng. Sau cùng là xuất ra kết quả hình ảnh với tên định dạng là <tên file ảnh gốc>_<tên chức năng thực hiện>.jpg

IV. Mô tả chi tiết các hàm

Ở mục này, ta sẽ lần lượt đi mô tả chi tiết các hàm xử lý chính của chương trình, cụ thể ta sẽ nói về các dữ liệu đầu vào, kết quả đầu ra và công dụng của các hàm.

Bảng mô tả chi tiết của các hàm của chương trình.





Tên hàm	Các biến đầu vào	Kết quả đầu ra	Công dụng
Read Image	<i>path</i> : là biến ở dạng chuỗi chứa tên của bức ảnh / đường dẫn của ảnh mà ta muốn đọc ảnh	Ảnh kết quả được lưu ở dạng kiểu dữ liệu mảng numpy	Dùng để chuyển hình ảnh cần xử lý về các dạng mảng numpy nhằm phục vụ các hàm ở phía sau
Show	<i>image</i> : tham số ảnh	Hiển thị ảnh ra màn hình	Hiển thị kết quả của chương

Image	truyền vào ở dạng mảng numpy.	máy tính kích thước ảnh.	trình chạy được ra màn hình.
SaveImage	<i>img_arr</i> : mảng ảnh kết quả, <i>format</i> : tên chức năng đã xử lý, <i>name</i> : tên gốc của ảnh	Ảnh được lưu ở định dạng <tên file ảnh gốc>_<tên chức năng thực hiện>.jpg	Lưu ảnh kết quả vào máy.
RGB2Gray	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh xám kích thước m*n	Chuyển đổi ảnh màu qua ảnh xám.
Brightness	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy <i>C</i> : đại lượng hiệu chỉnh độ sáng	Ảnh đã được thay đổi độ sáng tương ứng	Thay đổi độ sáng của ảnh
Contrast	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy <i>C</i> : đại lượng hiệu chỉnh độ tương phản	Ảnh đã thay đổi độ tương phản	Thay đổi độ tương phản của ảnh
RGB2Sep	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh ở dạng sepian kích thước m*n*3	Chuyển ảnh về chế độ Sepian
CircleCrop	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh được xén thành khuôn hình tròn, giống ảnh đại diện	Xén ảnh theo hình tròn
CropCenter	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh bị xén đi, chỉ còn khung hình ở trung tâm ảnh gốc	Xén hình theo trung tâm
flipVertical/ flip Horizon	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh kết quả được lấy đối xứng qua trục hoành/tung trung tâm ảnh	Lật ảnh dọc/ngang
BlurImage	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh đã được làm mờ	Tạo Gaussian kernel sau đó thực hiện convolution để làm mờ ảnh
Sharpen	<i>img</i> : tham số ảnh truyền vào ở dạng mảng numpy	Ảnh đã được nâng độ phân giải.	Nâng độ phân giải ảnh thông qua Gaussian kernel


Main	Không có	Chạy thực thi chương trình	Cho người dùng chọn ảnh và các thông số + pp xử lý và thực hiện

V. Khảo sát kết quả thực thi của chương trình

Ở phần này, ta sẽ thực hiện quá trình chạy demo của hệ thống.với ảnh đầu vào là lena.png, ta có:

Tác Vụ	Ảnh gốc	Ảnh qua xử lý
Hiệu chỉnh độ sáng ảnh Giá trị truyền vào: -12		
Hiệu chỉnh độ tương phản Giá trị truyền vào: 1.5		

Lật ảnh ngang		
Lật ảnh dọc		
Chuyển đổi ảnh xám		
Chuyển đổi ảnh Sepia		

<p>Làm mờ ảnh với kích thước kernel là 5x5 ($k = 5$) Giá trị sigma = 1.5</p>		
<p>Làm sắc nét ảnh với kích thước kernel là 5x5 ($k = 5$) Giá trị sigma = 1.5</p>		
<p>Cắt hình theo trung tâm</p>		
<p>Cắt hình theo khuôn tròn</p>		

VI. Nhận xét chung và kết luận

Như vậy, sau khi chạy toàn bộ kết quả, ta thấy mô hình hoạt động trơn tru và nhanh nhẹn. Ngoài ra, hàm làm mờ ảnh khi khởi tạo kernel ở hàm Gaussssian_filter với $k = 5$ thì chỉ mất 5.3s để ra kết quả, còn khi $k = 3$ thì mất 7.4s để ra được kết quả.

⇒ Tất cả các tác vụ trên đều hoạt động trơn tru và hiệu quả 100% theo đúng yêu cầu của cô giao.

Hình ảnh các kết quả đã làm được khi truyền ảnh vào cho chương trình xử lý.



Trong đó, venv là môi trường ảo, lena.png là ảnh gốc truyền vào, và các ảnh đằng sau là kết quả xử lý tương tự.

Lưu ý: Khi chạy hàm main ở chế độ 0 (thực thi toàn bộ các hàm) trên jupyternotebook thì các kết quả hiển thị hình ảnh thông qua hàm ShowImage ra console chỉ hiển thị kết quả của phần xử lý cuối cùng. Vì thế, nếu muốn hiển thị tốt hơn thì ta có thể chạy từng tính năng một. Đồng thời, khi chạy hàm main ở chế độ 0 thì các hàm sẽ chỉ thông báo ra kết quả chứ không in ra ảnh, nhưng ảnh kết quả được lưu như hình minh họa ở trên.

VII. Nguồn tài liệu tham khảo

1. <https://dyclassroom.com/image-processing-project/how-to-convert-a-color-image-into-sepia-image>
2. <https://www.baeldung.com/cs/convert-rgb-to-grayscale>
3. https://en.wikipedia.org/wiki/Gaussian_blur
4. https://scipy-lectures.org/advanced/image_processing/auto_examples/plot_sharpen.html
5. Tài liệu của cô Uyên và thầy Hoàng cung cấp