

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN - ĐHQG TP.HCM

VNUHCM - UNIVERSITY OF SCIENCE

KHOA CÔNG NGHỆ THÔNG TIN

Image Processing Project

Họ và Tên: Thành Thiện Nhân

Mã số sinh viên: 21127535

1) Các chức năng đã làm được:

Trong đồ án này, em đã học được thêm một số thao tác với ảnh màu RGB, các chức năng mà em đã làm được là:

- Thay đổi độ sáng cho ảnh
- Thay đổi độ tương phản
- Lật ảnh (Người dùng chọn kiểu lật ngang hoặc lật dọc)
- Chuyển đổi ảnh RGB thành ảnh xám/sepia (Người dùng chọn một trong hai cách)
- Làm mờ/sắc nét ảnh (Người dùng chọn một trong hai cách)
- Cắt ảnh theo kích thước (Cắt ở trung tâm)
- Cắt ảnh theo khung hình tròn
- Cắt ảnh theo khung hai hình ellip

Trong quá trình dùng ảnh để thao tác, em đã nhận thấy rằng một số ảnh định dạng ‘png’ có đến 4 kênh, gây lỗi trong quá trình reshape ảnh đầu ra (cụ thể là ở chức năng chuyển đổi ảnh sepia với ma trận chỉ cần 3 kênh màu). Kênh ảnh thứ tư này chính là kênh alpha đại diện cho độ mờ của mỗi pixel, có giá trị nhỏ nhất là 0 thể hiện độ trong suốt hoàn toàn, và giá trị lớn nhất là 255 thể hiện sự không trong suốt của ảnh. Với yêu cầu của đồ án em sẽ chỉ thao tác với các ảnh có ba kênh là RGB với dòng lệnh ‘`img_input = img_input.convert("RGB")`’ để chuyển thành ảnh chỉ có 3 kênh.

2) Ý tưởng thực hiện và mô tả các hàm chức năng:

a) Thay đổi độ sáng của ảnh:

Ở từng điểm ảnh, giá trị từ [0,255] thể hiện độ sáng của từng kênh màu R (red), G (green), B (blue), với giá trị 0 thì điểm ảnh sẽ hoàn toàn đen và giá trị 255 điểm ảnh sẽ sáng thành màu trắng.

Để có thể tăng độ sáng của ảnh, em sẽ cộng một giá trị bias (trong đồ án này em đặt bias là 50) để cộng vào giá trị từng điểm ảnh nhằm tăng độ sáng tổng thể cho ảnh.

Hàm chức năng: ***change_brightness(img_input,bias=50)***: Hàm này sẽ nhận ma trận ảnh đầu vào và một giá trị bias để tăng độ sáng (bias lớn hơn 0 sẽ tăng độ sáng ảnh và bias nhỏ hơn 0 sẽ giảm độ sáng ảnh). Đầu tiên em đặt ma trận `img_output` cùng kích cỡ với ma trận `img_input` nhưng được đưa về dạng 'uint16' nhằm tránh việc tràn số khi giá trị sau khi cộng bias vượt quá 255. Tiếp theo là bước tiến hành cộng giá trị bias vào từng điểm ảnh của `img_output`. Sau đó, em dùng hàm `np.clip` để giới hạn biên giá trị về [0,255]. Cuối cùng hàm sẽ trả về ma trận sau khi cộng bias vào ở dạng 'uint8' có giá trị từ 0 đến 255.

b) Thay đổi độ tương phản của ảnh:

Để tăng độ tương phản của ảnh, em cần phải tăng khoảng cách giữa các giá trị màu để hiển thị rõ độ chênh lệch sáng-tối, màu của đối tượng này với đối tượng khác.

Sau khi tìm hiểu, em đã tìm được một trong những công thức để thay đổi độ tương phản của ảnh là:

$$factor * (giá\ trị\ điểm\ ảnh - 128) + 128$$

Với factor là giá trị thể hiện sự điều chỉnh độ tương phản cho ảnh dựa vào giá trị alpha (trong đồ án này em đặt alpha=50) như sau:

$$factor = (259 * (alpha+255)) / (255 * (259 - alpha))$$

Hàm chức năng: ***change_contrast(img_input,alpha=50)***: Đầu tiên em sẽ đặt ma trận `img_output` cùng kích cỡ với ma trận `img_input` nhưng được đổi thành dạng 'float32' để tránh việc bị tràn số. Sau đó em sẽ tính giá trị factor dựa vào giá trị alpha (giá trị alpha lớn hơn 0 sẽ tăng độ tương phản và alpha nhỏ hơn 0 sẽ giảm độ tương phản), rồi dùng giá trị factor này để thay đổi độ tương phản cho ảnh như hai công thức trên. Cuối cùng sẽ trả về ma trận `img_output` ở dạng 'uint8'.

c) Lật ảnh:

Để lật ảnh lần lượt theo chiều ngang và chiều dọc, em sẽ dùng hàm flip của numpy để đảo ma trận tương ứng với trục hoành x là 'axis=1', trục tung y là 'axis=0'.

Hàm chức năng: **flip_image(img_input,style)**: Nếu style được chọn là 'horizontal' thì ma trận sẽ được lật theo chiều ngang, còn nếu style là 'vertical' thì ma trận sẽ được đảo theo chiều dọc.

d) Chuyển ảnh về ảnh xám (grayscale) hoặc sepia:

Để chuyển một bức ảnh RGB thành một bức ảnh xám, em sẽ dựa vào công thức:
 $0.299*Red + 0.587*Green + 0.114*Blue$

Để chuyển một bức ảnh RGB thành một bức ảnh sepia, em sẽ dùng công thức biến đổi theo từng kênh màu như sau:

$$Sepia_r = (Red * 0.393) + (Green * 0.769) + (Blue * 0.189)$$

$$Sepia_g = (Red * 0.349) + (Green * 0.686) + (Blue * 0.168)$$

$$Sepia_b = (Red * 0.272) + (Green * 0.534) + (Blue * 0.131)$$

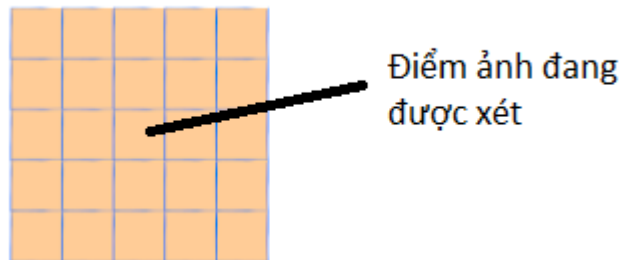
Hàm chức năng: **grayscale_sepia(img_input,style)**: Đầu tiên em sẽ tạo một ma trận temp đã được reshape từ 3 chiều thành ma trận 2 chiều. Khi đó em sẽ có ma trận có 3 cột với mục đích là lấy tất cả dòng ở từng cột lần lượt là r, g, b để có thể chuyển đổi ảnh như công thức đã liệt kê.

Với style 'grayscale', sau khi biến đổi theo công thức, em sẽ tiến hành reshape ma trận img_output thành ma trận 3 chiều với chỉ 1 kênh màu xám, nhưng để in ra ảnh vẫn theo format RGB 3 kênh thì em đã dùng hàm 'np.dstack' để chồng ma trận, nhằm tạo ra ma trận 3 chiều sao cho mỗi điểm ảnh có giá trị xám giống nhau cho cả ba kênh.

Tương tự với style 'sepia' thì sau khi biến đổi theo công thức em sẽ dùng hàm 'np.dstack' để chồng ma trận theo 3 kênh để thể hiện độ sáng gốc của từng kênh màu Red, Green, Blue sau khi được chuyển sang dạng sepia.

e) Làm mờ ảnh/Làm sắc nét ảnh:

Để làm mờ ảnh, em sẽ đặt một giá trị `kernel_size` là một ma trận vuông (trong đồ án này em đặt `kernel_size=5` để làm ma trận 5x5) nhằm tính giá trị trung bình của các pixel trong ảnh, lưu ý rằng `kernel_size` nên được đặt là số lẻ để điểm ảnh đang xét có thể nằm ở chính giữa kernel. Em đã đặt thêm tham số `padding` để quy định việc thêm đệm vào ảnh đầu vào để xử lý các pixel ở mép của ảnh.



Hình ảnh minh họa cho kernel với kích thước 5x5

Hàm chức năng: **`blur_sharpen(img_input, style, kernel_size=5)`**: Đầu tiên em sẽ quy định đệm `padding` là: $(kernel_size) - 1 // 2$, và tạo một ma trận không bằng với kích thước của `img_input`.

Với style là 'blur', đối với mỗi pixel ở vị trí (i,j), hàm sẽ tính giá trị trung bình của các pixel trong kernel xung quanh pixel đó. Kernel được xác định bằng cách lấy các pixel từ hàng ' $\max(0, i-padding)$ ' đến ' $\min(i+padding+1, row)$ ', và lấy từ cột ' $\max(0, j-padding)$ ' đến ' $\min(j+padding+1, column)$ '. Giá trị trung bình được dùng từ hàm '`np.mean`' sẽ được gán cho pixel tương ứng trong ảnh đầu ra. Ảnh sau khi trả về sẽ có hiệu ứng được làm mờ, với `kernel_size` càng lớn thì ảnh đầu ra sẽ càng mờ hơn.

Với style là 'sharpen', hàm sẽ tính giá trị trung bình của các pixel trong kernel tương tự như cách làm mờ ảnh. Sau đó, hàm tính toán giá trị 'sharpened' cho pixel đó bằng cách lấy giá trị ban đầu của pixel và cộng với hiệu của giá trị ban đầu và giá trị trung bình trong kernel. Khi đó giá trị 'sharpen' sẽ được gán cho pixel tương ứng trong ảnh đầu ra nhằm tạo hiệu ứng ảnh đầu ra sắc nét hơn ảnh gốc

f) *Cắt ảnh theo kích thước (cắt ở trung tâm):*

Trong đồ án này em thực hiện cắt ảnh ảnh đầu vào thành một ảnh con có kích thước bằng một nửa kích thước của ảnh gốc.

Hàm chức năng: ***crop_center(img_input)***: Đầu tiên em lấy một nửa chiều rộng là 'crop_width' và một nửa chiều cao là 'crop_height' từ việc thực hiện phép chia floor division cho 2 (floor division lấy số nguyên ở cạnh bên trái của kết quả). Sau đó tính các vị trí bắt đầu để cắt ảnh với 'start_x' là vị trí cột bắt đầu (tính từ trái sang) và 'start_y' là vị trí dòng bắt đầu (tính từ trên xuống) nhằm cắt ảnh con có điểm trung tâm cũng là trung tâm ảnh gốc. Cuối cùng, em tiến hành cắt ảnh đầu vào tính từ vị trí bắt đầu với kích thước chiều rộng 'crop_width' và chiều cao 'crop_height' đã được tính trước, rồi trả về ma trận ảnh con sau khi được cắt.

g) *Cắt ảnh theo khung hình tròn:*

$$\text{Phương trình đường tròn: } (X - a)^2 + (Y - b)^2 = R^2$$

Để tính bán kính R nhằm tạo ra đường tròn, em sẽ lấy căn bậc hai của vế trái phương trình đường tròn. Sau đó tạo một 'mask' để tạo ma trận ảnh đầu ra sao cho các điểm nằm bên trong đường tròn sẽ vẫn hiển thị như ảnh gốc, nhưng bên ngoài đường tròn sẽ là màu đen.

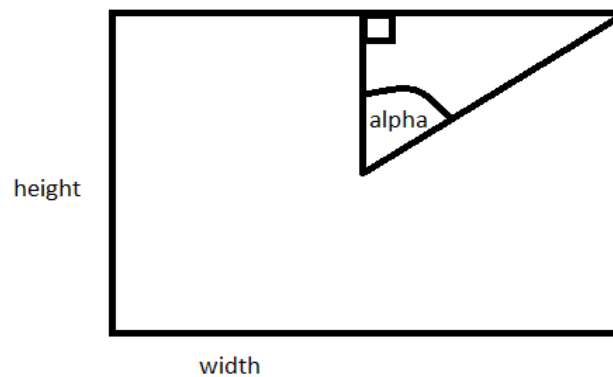
Hàm chức năng: ***crop_circle(img_input)***: Đầu tiên em tiến hành tính giá trị trung tâm của ảnh 'center_x' và 'center_y'. Em cũng tính trước bán kính hình tròn 'radius' với giá trị là 'min(center_x, center_y)' để đường tròn nằm gọn trong ảnh trong trường hợp ảnh là hình chữ nhật. Tiếp theo, em tính các giá trị 'x_indices' và 'y_indices' để tạo nên lưới tọa độ, rồi tính khoảng cách 'distance' là căn bậc hai của vế trái phương trình đường tròn. Sau khi có được 'distance' và 'radius', em tạo một 'mask' bằng cách chọn các điểm ảnh nằm trong hoặc trên đường tròn bán kính 'radius' (trả về False nếu nằm ngoài bán kính). Tiếp theo, em tạo một ma trận img_output là một ma trận không với kích thước như ảnh gốc, việc tạo ma trận không sẽ ban đầu tạo ra một bức ảnh đen để áp dụng mask và giữ lại phần tương ứng của ảnh gốc mà ở bên trong hình tròn. Cuối cùng, em trả về ma trận ảnh đầu ra với khung hình tròn và bên ngoài hình tròn là nền đen.

h) Phần nâng cao: Cắt ảnh theo khung hai hình ellip chéo nhau

Phương trình ellip: $(X^2/a^2) + (Y^2/b^2) = 1$

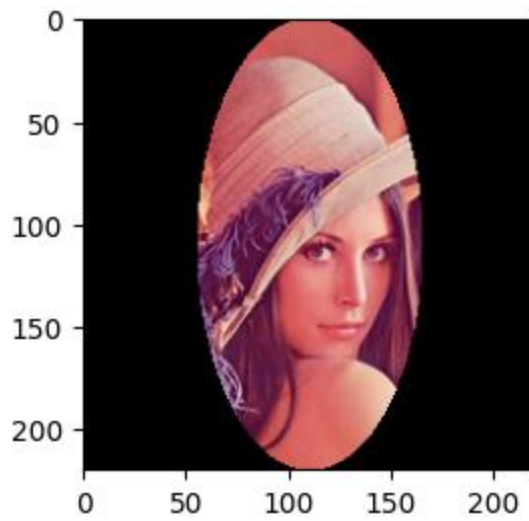
Về phần này em đã có thể tạo ra 2 ellip sao cho chiều dài sẽ hướng theo đường chéo của ảnh, nhưng em vẫn chưa thể xác định kích thước sao cho hình ellip sẽ chạm vào biên của ảnh ở 8 điểm như ảnh gốc, nên trong đồ án này em sẽ chỉ tạo ellip với kích thước là tính một nửa chiều dài và chiều rộng, rồi lấy min để ellip nằm gọn trong ảnh (tương tự như phần ‘Cắt ảnh theo khung hình tròn’). Về ý tưởng, em sẽ tạo ra hai hình ellip với phương trình ellip là như nhau, nhưng mỗi ellip sẽ được xoay theo một góc alpha để có thể hướng về 4 góc của ảnh.

Sau một thời gian tìm hiểu, em tính được góc alpha cần tìm là giá trị $\tan^{-1}(\tan_alpha)$ của tam giác sau:

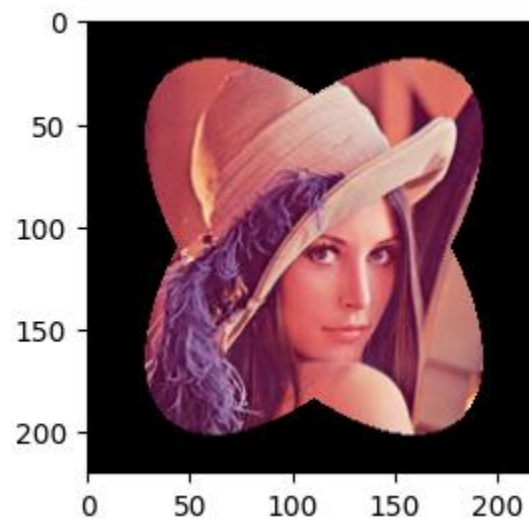


Hình minh họa cho một bức ảnh với chiều cao height và độ rộng width

Trong đó, \tan_alpha chính là giá trị tan của góc alpha, cụ thể hơn là $(width/2) / (height/2)$ theo hệ thức lượng trong tam giác vuông. Để tính $\tan^{-1}(\tan_alpha)$ thì em sẽ dùng hàm ‘inverse_tan()’ có chức năng tính \tan^{-1} (hay còn gọi là arctan).



*Hai ellip có cùng phương trình nên người dùng
sẽ chỉ nhìn thấy một hình ellip trong ảnh*



Hai ellip sau khi xoay theo góc 'alpha' và '-alpha'

Hàm chức năng: ***crop_ellipses(img_input)***: Đầu tiên em tính các giá trị ‘center_x’, ‘center_y’ tương tự với bài toán cắt khung đường tròn, cùng với bán kính theo trục x và y của hình ellip ‘radius_x’ và ‘radius_y’ với ‘radius_x’ dài bằng một nửa ‘radius_y’.

Tiếp theo, em tính ‘x_indices’ và ‘y_indices’ nhằm tạo lưới tọa độ với hàm ‘np.indices’.

Em tiến hành tính góc xoay alpha với đơn vị là radian bằng cách lấy lấy giá trị tan là ‘tan_alpha’ được tính như ý tưởng được đề cập bên trên, sau đó sẽ tìm góc alpha bằng cách lấy hàm ‘inverse_tan’.

Sau đó, em tính phương trình đường ellip của hai hình ellip. Để xoay ellip thì em sẽ dùng công thức theo dạng:

$$\frac{[X \cos(\alpha) + Y \sin(\alpha)]^2}{a^2} + \frac{[Y \cos(\alpha) + X \sin(\alpha)]^2}{b^2}$$

Với ellip thứ nhất sẽ xoay theo góc ‘alpha’, Với ellip thứ hai sẽ xoay theo góc alpha nhưng theo chiều ngược lại là ‘-alpha’.

Tương tự như bài toán với khung tròn, em tạo một ‘mask’ chọn các điểm nằm trong từng vùng ellip bằng cách dùng toán tử ‘|’ như là phép or để xác định giá trị của từng điểm trong hình có nằm trong hay nằm trên đường ellip thứ nhất hoặc ellip thứ hai hay không. Sau đó em tạo một ma trận img_output là ma trận không, giữ lại phần bên trong ellip (tương ứng với các giá trị True trong mask), và trả về ma trận ảnh đầu ra như đã làm ở bài toán cắt ảnh theo khung tròn.

i) Một số hàm khác:

print_menu(): In ra menu với các chức năng cho người dùng

input_function_num(): Yêu cầu người dùng nhập số từ 0 đến 8 để chọn chức năng

input_1_or_2(Option1,Option2): Yêu cầu người dùng nhập 1 hoặc 2 để lựa chọn cho các chức năng có 2 trường hợp thực hiện.

inverse_tan(x): Tính arctan

plot_image(Before,After): Giúp biểu diễn trực quan ảnh gốc và ảnh sau khi thực hiện chức năng

run_one_function(img_input,function_num): Chạy một trong 8 chức năng, có yêu cầu người dùng nhập 1 hoặc 2 ở các chức năng có 2 trường hợp thực hiện

save_one_image(img_name,img_output,function_num,choice): Lưu một ảnh theo định dạng

run_all_function(img_input): Chạy tất cả các chức năng, mỗi chức năng chạy tất cả các trường hợp (tổng cộng có 11 chức năng được chạy).

save_all_image(img_name,img_output): Lưu tất cả các ảnh ở 11 chức năng được chạy này.

Hàm *main()*: Đầu tiên sẽ yêu cầu người dùng nhập tên ảnh. Tiến hành mở ảnh, chuyển về dạng ảnh màu 'RGB' và biến thành ma trận với hàm 'np.asarray()'. Sau đó sẽ in menu, yêu cầu người dùng chọn chức năng rồi chương trình sẽ trả ra kết quả mong muốn.

3) Hình ảnh kết quả với từng chức năng:

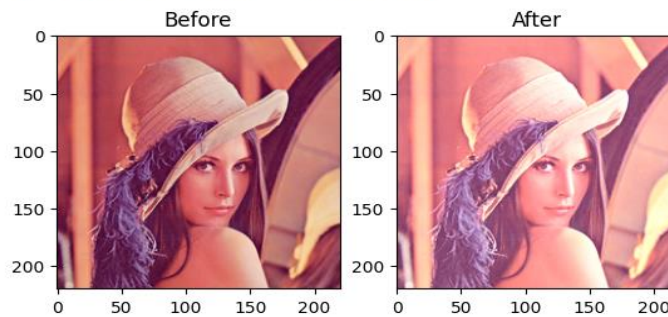
Cho ảnh lena.png như sau:



lena.png

Khi chọn chức năng 1:

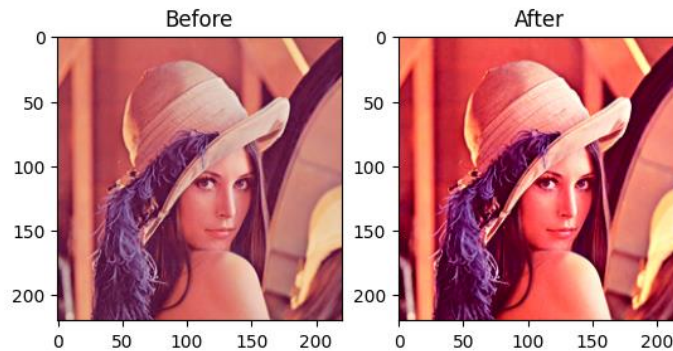
```
, Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 1
```



Ảnh được thay đổi độ sáng

Khi chọn chức năng 2:

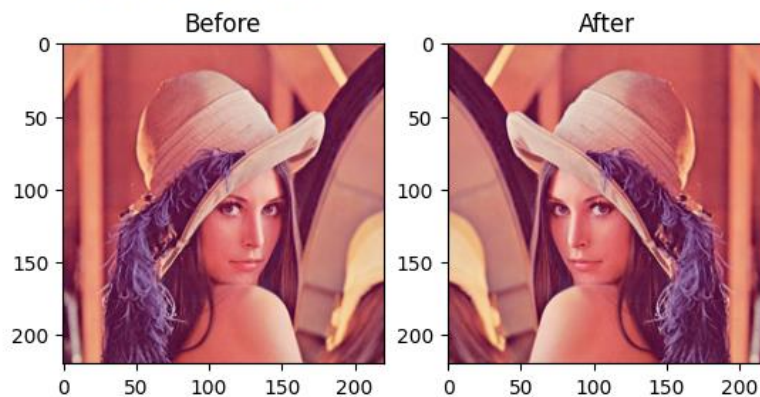
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 2
```



Ảnh được thay đổi độ tương phản

Khi chọn chức năng 3, với trường hợp lật ngang:

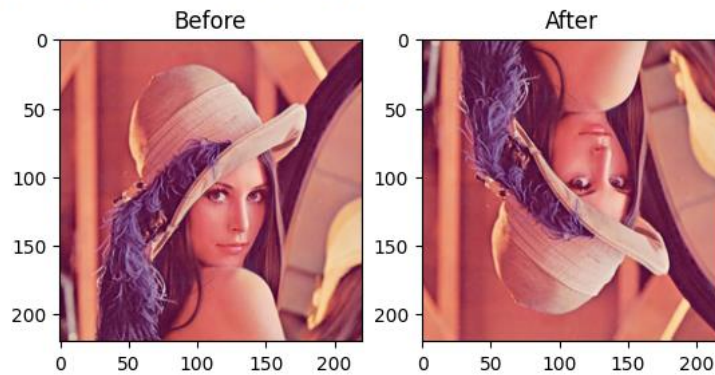
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 3
1. Flip horizontal
2. Flip vertical
Please input your choice (1 or 2): 1
```



Ảnh được lật ngang

Khi chọn chức năng 3, với trường hợp lật dọc:

```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 3
1. Flip horizontal
2. Flip vertical
Please input your choice (1 or 2): 2
```



Ảnh được lật dọc

Khi chọn chức năng 4, với trường hợp chuyển thành ảnh xám:

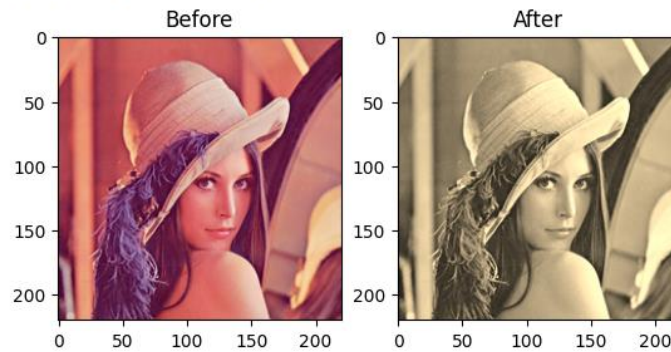
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 4
1. Change to grayscale image
2. Change to sepia image
Please input your choice (1 or 2): 1
```



Ảnh được chuyển thành ảnh xám

Khi chọn chức năng 4, với trường hợp chuyển thành ảnh sepia:

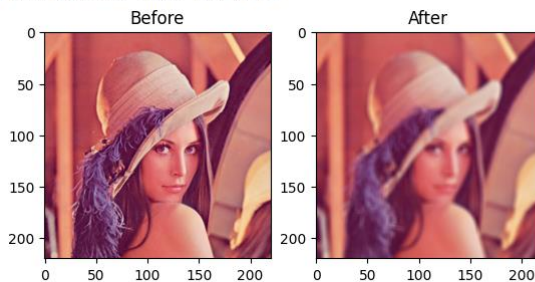
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 4
1. Change to grayscale image
2. Change to sepia image
Please input your choice (1 or 2): 2
```



Ảnh được chuyển thành ảnh sepia

Khi chọn chức năng 5, với trường hợp làm mờ ảnh:

```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 5
1. Blur image
2. Sharpen image
Please input your choice (1 or 2): 1
```



✓ 7 giây hoàn thành lúc 21:45

Ảnh được làm mờ, trong khoảng thời gian 7 giây

Khi chọn chức năng 5, với trường hợp làm sắc nét ảnh:

```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 5
1. Blur image
2. Sharpen image
Please input your choice (1 or 2): 2
```

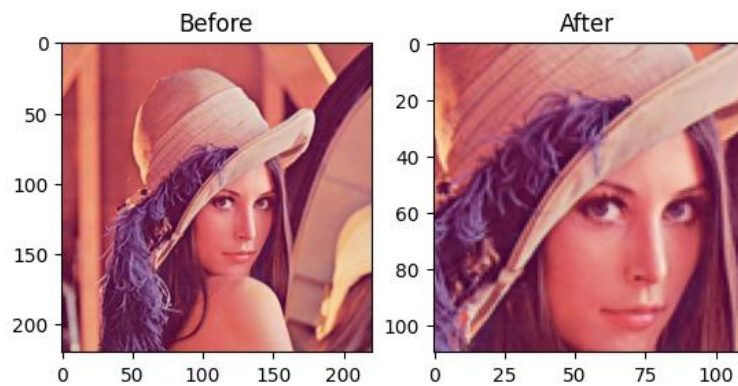


✓ 10 giây hoàn thành lúc 21:46

Ảnh được làm sắc nét hơn, trong khoảng thời gian 10 giây

Khi chọn chức năng 6:

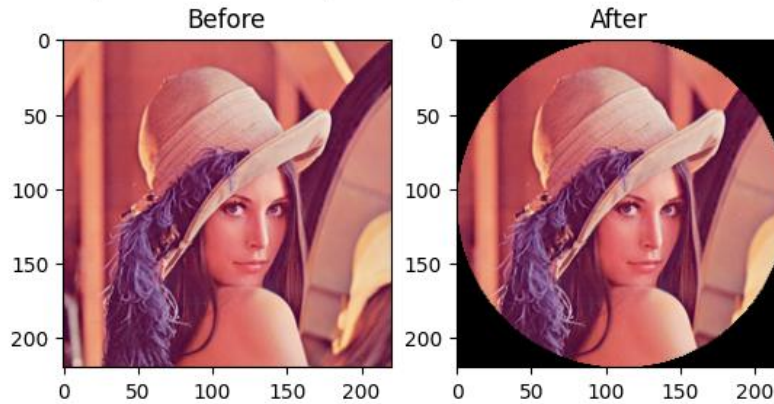
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 6
```



Ảnh được cắt ở trung tâm, có kích thước bằng một nửa ảnh gốc

Khi chọn chức năng 7:

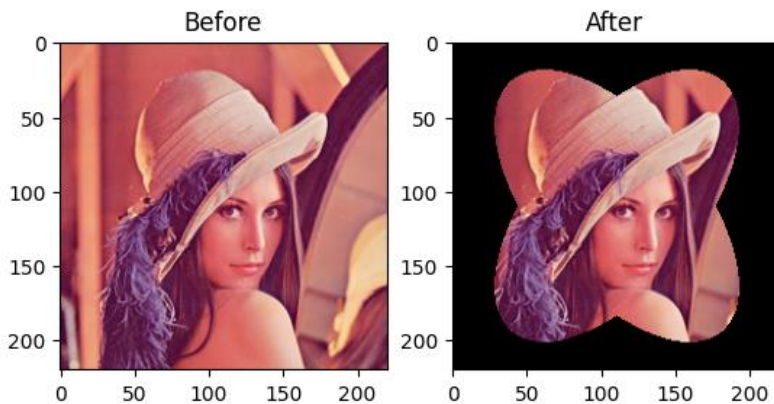
```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 7
```



Ảnh được cắt theo khung hình tròn

Khi chọn chức năng 8:

```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 8
```



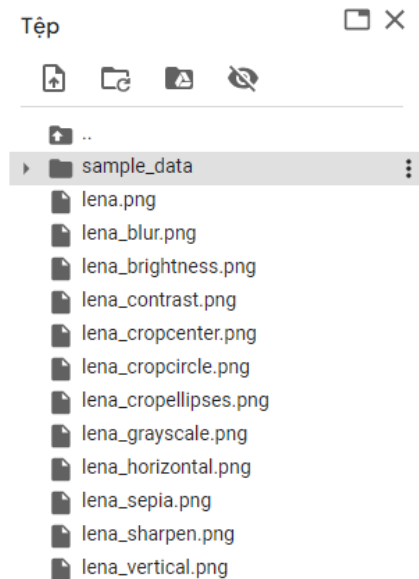
Ảnh được cắt theo khung là hai hình ellip chéo nhau

Khi chạy một chức năng thì chương trình sẽ lưu hình ảnh đầu ra theo định dạng ứng với chức năng đó (ví dụ nhập ảnh là 'cat.png' và chọn chức năng làm mờ thì ảnh đầu ra có tên là 'cat_blur.png').

Trong trường hợp cuối cùng là trường hợp 0 (chạy tất cả các chức năng), em sẽ lưu tất cả các ảnh đầu ra của từng chức năng:

```
Please input the name of the image (for example: image.jpg): lena.png
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 0
Complete running all functions! Please check the output files.
```

Màn hình hiển thị



Lưu về thêm 11 file ảnh đầu ra

Tiếp theo, em sẽ tiến hành chạy chức năng làm mờ và làm sắc nét ảnh với kích thước 512x512 để kiểm tra thời gian hoàn thành hai chức năng này.

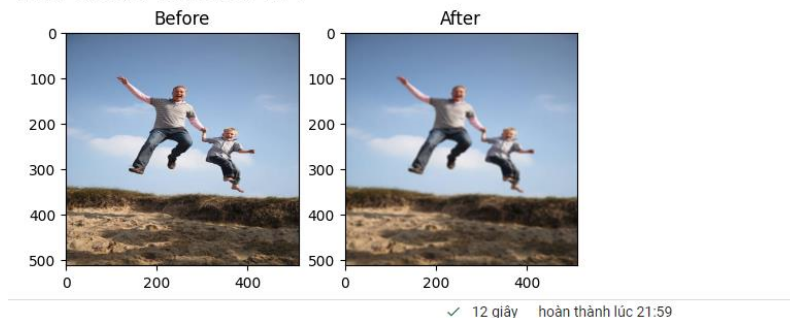
Cho ảnh image.jpg với kích thước 512x512:



Ảnh image.jpg được chuyển về kích thước 512 x 512

Khi dùng chức năng làm mờ ảnh:

```
Please input the name of the image (for example: image.jpg): image.jpg
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 5
1. Blur image
2. Sharpen image
Please input your choice (1 or 2): 1
```



Ảnh được làm mờ trong khoảng thời gian 12 giây (tính cả thời gian nhập liệu)

Khi dùng chức năng làm sắc nét ảnh:

```
Please input the name of the image (for example: image.jpg): image.jpg
0. Run All Below Functions
1. Change Brightness
2. Change Contrast
3. Flip Image Horizontal/Vertical
4. Change Image into Grayscale/Sepia
5. Blur/Sharpen the Image
6. Crop Image (at center)
7. Crop Image in Circle
8. Crop Image in Two Ellipses
Please input function number (from 0 to 8): 5
1. Blur image
2. Sharpen image
Please input your choice (1 or 2): 2
```



✓ 18 giây hoàn thành lúc 22:01

Ảnh được làm sắc nét trong khoảng thời gian 18 giây (tính cả thời gian nhập liệu)

4) Một số tài liệu tham khảo:

Tìm hiểu về kênh màu thứ tư – kênh alpha:

<https://www.makeuseof.com/tag/alpha-channel-images-mean/>

Cách loại bỏ kênh alpha khỏi ảnh:

<https://stackoverflow.com/questions/35902302/discarding-alpha-channel-from-images-stored-as-numpy-arrays>

Công thức tham khảo để thay đổi độ tương phản của ảnh:

<https://stackoverflow.com/questions/42045362/change-contrast-of-image-in-pil>

Cách chuyển ảnh RGB thành grayscale:

https://support.ptc.com/help/mathcad/r9.0/en/index.html#page/PTC_Mathcad_Help/example_grayscale_and_color_in_images.html

Cách chuyển ảnh RGB thành sepia:

<https://dyclassroom.com/image-processing-project/how-to-convert-a-color-image-into-sepia-image>

Tìm hiểu hàm numpy.flip():

<https://www.geeksforgeeks.org/numpy-flip-python/>

Tìm hiểu hàm numpy.dstack():

<https://www.geeksforgeeks.org/python-numpy-dstack-method/>

Ý tưởng làm mờ ảnh:

<https://www.educative.io/answers/how-to-blur-an-image-using-a-mean-filter>

Trang web để thay đổi kích cỡ ảnh thành 512x512 với mục đích kiểm tra thời gian thực hiện chức năng:

<https://compressjpeg.online/resize-image-to-512x512>