

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ

Báo Cáo Bài Thực Hành – Lab 01

Giảng viên hướng dẫn

Thầy Lý Quốc Ngọc

Thầy Nguyễn Mạnh Hùng

Sinh viên thực hiện

Võ Nguyễn Hoàng Kim

21127090

PHỤ LỤC

I. BẢNG TỰ ĐÁNH GIÁ	3
II. PHÂN TÍCH – SO SÁNH KẾT QUẢ.....	3
1. Color transformation	4
a. Linear mapping	4
b. Probability Density Function – based mapping	5
2. Geometry transformation.....	6
a. Scale down.....	6
b. Scale up	7
3. Image smoothing	8
a. Average filter	8
b. Gaussian filter	9
c. Median filter	10
III. TÀI LIỆU THAM KHẢO.....	11

I. BẢNG TỰ ĐÁNH GIÁ

Chức năng		Mức độ hoàn thành	Ghi chú
Color transformation	Linear mapping	100%	
	Probability Density Function – based mapping	100%	Sử dụng phương pháp “Histogram Equalization”
Geometry transformation	Scale down image	100%	
	Scale up image	100%	
Image Smoothing	Average filter	100%	Sử dụng bộ lọc 3x3
	Gaussian filter	100%	Sử dụng kernel 3x3
	Median filter	100%	Sử dụng bộ lọc 3x3

II. PHÂN TÍCH – SO SÁNH KẾT QUẢ

Lưu ý:

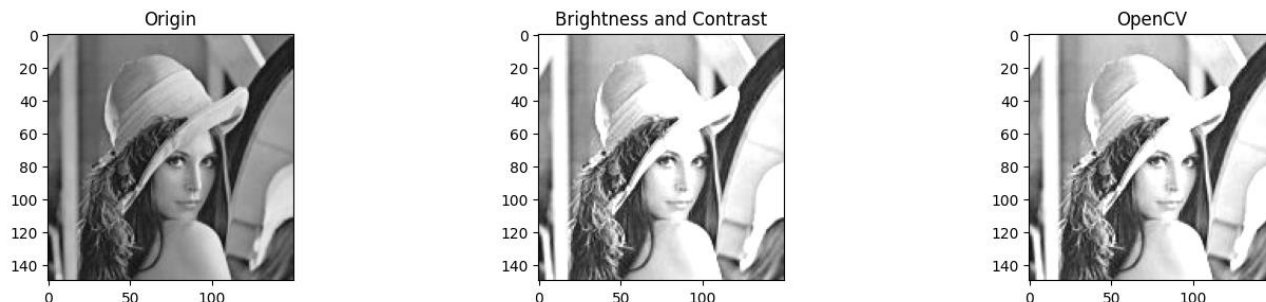
- Ảnh được đọc bằng đường dẫn tương đối bằng hàm cv2.imread(), dưới dạng cường độ xám (cv2.COLOR_GRAYSCALE).
- Các ảnh được sử dụng trong bài để so sánh, cài đặt, thực hiện thuật toán đều là ảnh ở cường độ xám.
- Dưới đây là ảnh mẫu được dùng để so sánh với các cài đặt (kích thước 150x150).
- Ảnh gốc được tải *tại đây*.



1. Color transformation

a. Linear mapping [1]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm cv2. convertScaleAbs (gọi là B):



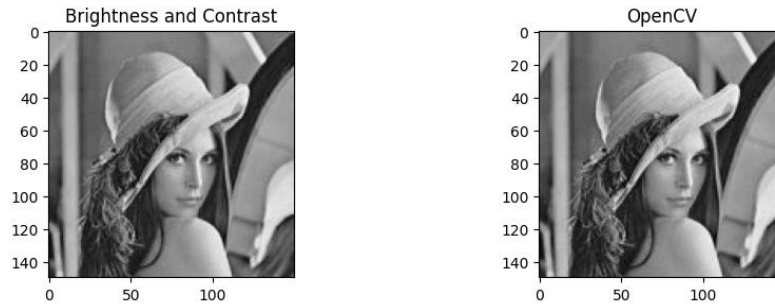
1.a.1 - $\alpha = 1.5$ và $\beta = 20$

Phương pháp:

- Thực hiện dựa trên hàm tuyến tính $g(x, y) = \alpha \cdot f(x, y) + \beta$ (gọi các hệ số là alpha và beta).
- Để tùy chỉnh độ sáng, ta thay đổi hệ số beta. Nếu beta là số dương, ảnh sẽ tăng độ sáng và ngược lại cho trường hợp số âm.
- Để tùy chỉnh độ tương phản, ta thay đổi hệ số alpha. Hệ số tương phản được dùng để điều chỉnh ảnh trong khoảng $[0, 1]$. Nếu alpha bé hơn 1, ảnh sẽ giảm độ tương phản và ngược lại đối với trường hợp $\alpha > 1$.

Nhận xét:

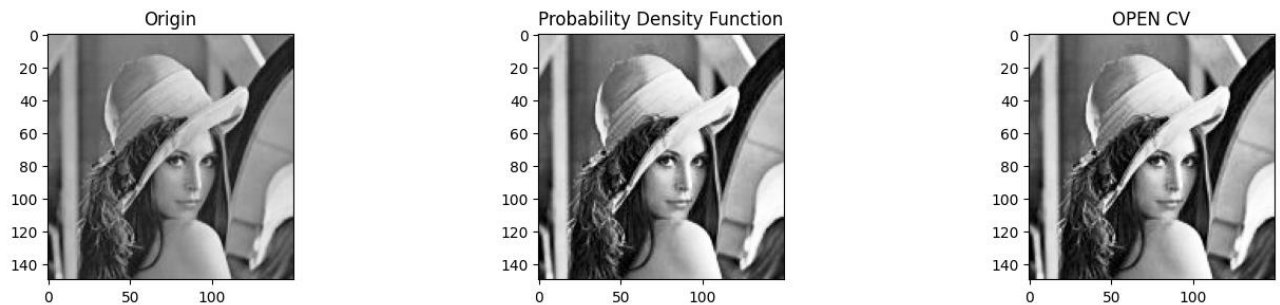
- Tổng quát
 - o Ảnh A đã đáp ứng được yêu cầu về việc điều chỉnh(tăng) độ sáng và độ tương phản so với ảnh gốc.
 - o Thuật toán được cài đặt cho ra kết quả phù hợp, vẫn giữ được nội dung của hình ảnh.
- So sánh với thư viện
 - o Có thể thấy, với cùng tham số alpha và beta như trên, nhưng ảnh A lại có màu sắc không được “mềm mại” như ảnh B. Có thể giải thích rằng, với việc điều chỉnh dựa theo hàm số tuyến tính ở trên, các pixel trong ảnh sẽ được thay đổi một cách tuyến tính, dựa theo giá trị mà nó tính toán được, dẫn đến việc tăng sẽ khá “thô cứng”. Còn đối với ảnh B, ta nhìn thấy có sự hài hòa giữa các pixel, không tạo chênh lệch rõ rệt giữa các pixel cận nhau, dẫn đến bức ảnh có sự mềm mại và dễ chịu khi nhìn.
 - o Tuy nhiên, sự khác biệt giữa ảnh A và B (tức thuật toán cài đặt và hàm thư viện) chỉ có thể thấy rõ rệt khi hệ số alpha và beta lớn. Trong trường hợp các hệ số nhỏ, giữa hai ảnh không có sự khác biệt nhiều (khó có thể phân biệt bằng mắt thường).



1.a.2 - $\alpha = 0.3$ và $\beta = 20$

b. Probability Density Function – based mapping [1]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm `cv2.equalizeHist` (gọi là B):



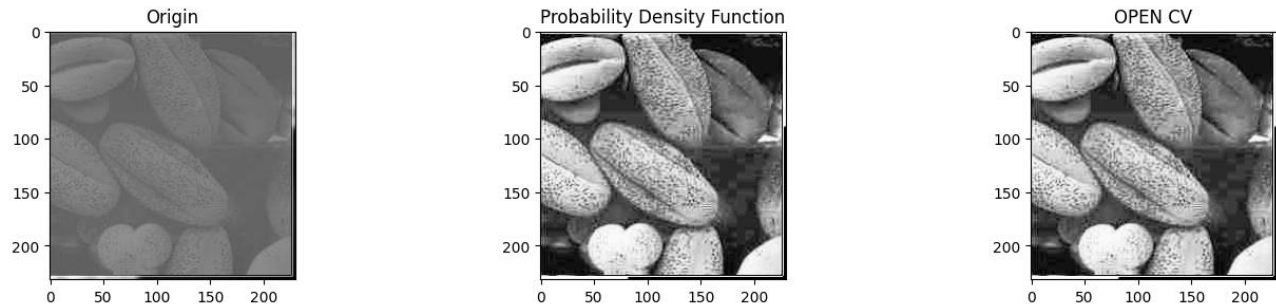
1.b.1 – PDF Example 1

Phương pháp

- Thuật toán Probability Density Function (gọi tắt là PDF) sử dụng các pixel lân cận để thay đổi giá trị của pixel đang xét.
- Nó được dùng cân bằng độ phân bố xác suất của các giá trị màu sắc, nhằm làm cho phân bố trở nên đồng đều. Cân bằng màu sắc có thể giúp tăng cường độ tương phản và làm cho hình ảnh trở nên mô tả đặc điểm màu sắc tốt hơn.
- Phương pháp được sử dụng trong trình bày này là “**Histogram Equalization**”

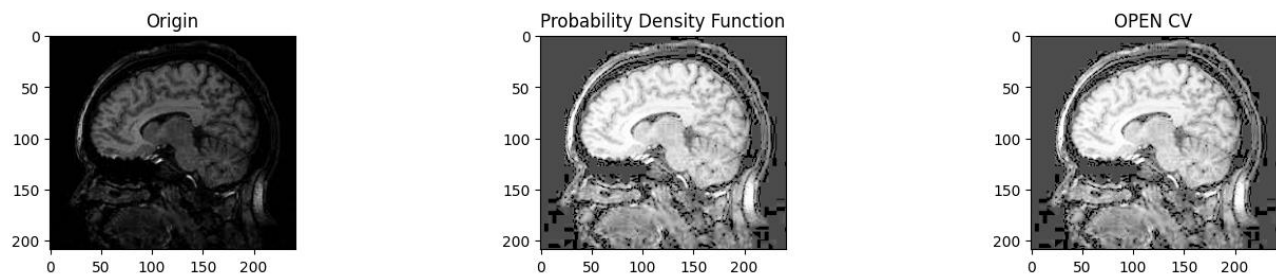
Nhận xét:

- Tổng quát:
 - Với hình ảnh Lenna trên, ta sẽ không thấy được độ hiệu quả mà PDF mang lại, dưới đây sẽ chèn thêm một kết quả của phép so sánh:

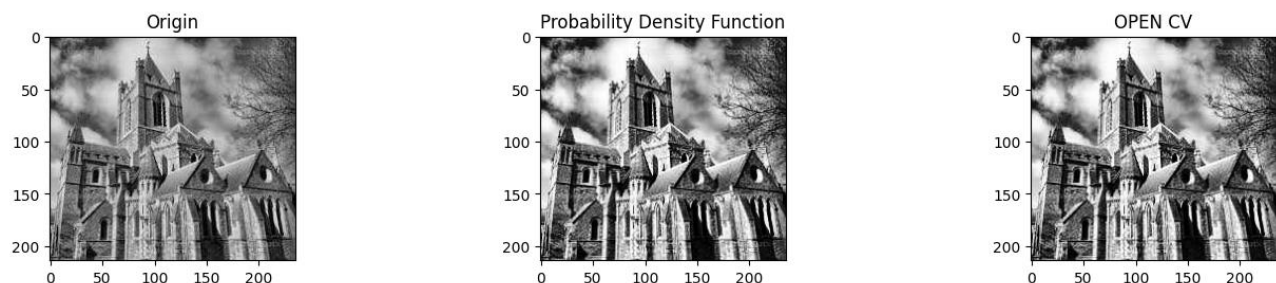


1.b.2 - PDF Example 2

- Có thể thấy, từ một bức ảnh xám không rõ chi tiết, với thuật toán PDF được cài đặt, ảnh đầu A ra đã có sự phân bố đồng đều các giá trị màu sắc, dẫn đến hình ảnh trở nên mô tả màu sắc hơn và thấy rõ các chi tiết hơn.
- So với thư viện
 - Nhìn chung, kết quả được trả ra giữa thuật toán PDF được cài đặt và hàm thư viện không có sự khác biệt, ở trường hợp này. Có thể đối chiếu thêm qua một số hình ảnh so sánh:



1.b.3 - PDF Example 3

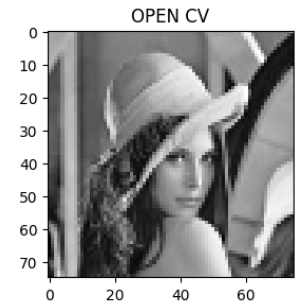
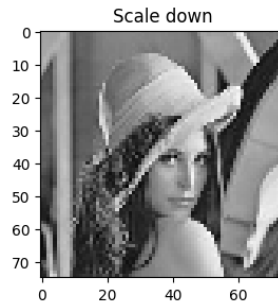
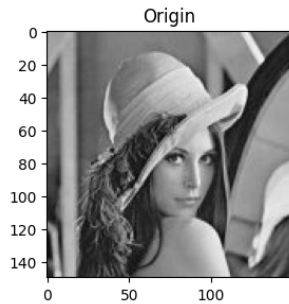


1.b.4 - PDF Example 4

2. Geometry transformation

a. Scale down [2]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm `cv2.resize` (gọi là B):



$$2.a.1 - f_x = 0.5 \text{ và } f_y = 0.5$$

Phương pháp

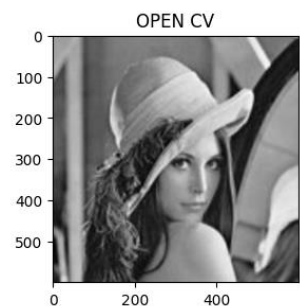
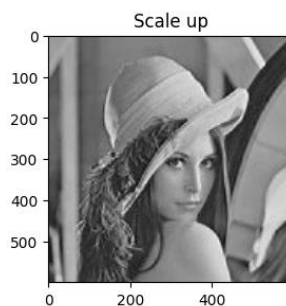
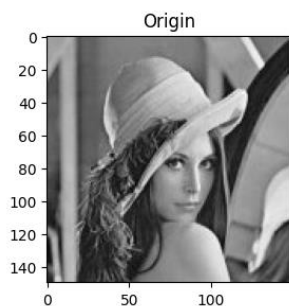
- Để thực hiện scale down ảnh, ta thực hiện giảm kích cỡ ảnh trên lưới tọa độ bằng cách nhân các điểm ảnh với hệ số f_x và f_y . (Pixel co-ordinate transformation)
- Tiếp đến, nội suy giá trị màu cho ảnh vừa được scale trên bằng phương pháp Linear interpolation (Brightness interpolation).

Nhận xét

- Tổng quát
 - Với kết quả của việc thu nhỏ ảnh (ảnh A), có thể thấy: kích thước đã được thu nhỏ (75x75), ảnh A vẫn giữ được nội dung của ảnh ban đầu, vẫn có thể thấy được các đường nét, chi tiết, thông tin, nội dung của ảnh gốc.
 - Tuy nhiên, chất lượng ảnh kết quả A không được tốt, có hiện tượng blocky khiến chất lượng ảnh giảm, ảnh không được “trơn tru” và mềm mại như ban đầu.
- So với thư viện
 - Tuy cùng giảm kích thước như nhau, tuy nhiên ảnh B có chất lượng tốt hơn, ít bị biến đổi nhiều, vẫn giữ được tính mềm mại.
 - Hiện tượng blocky xảy ra giữa A và B có thể thấy rõ, tuy nhiên hiện tượng này xảy ra với cường độ cao hơn ở ảnh A (thuật toán tự cài đặt) so với ảnh B (từ thư viện)

b. Scale up [2]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm `cv2.resize` (gọi là B):



$$2.b.1 - f_x = 4 \text{ và } f_y = 4$$

Phương pháp

- Để thực hiện scale up ảnh, ta thực hiện giảm kích cỡ ảnh trên lưới tọa độ bằng cách nhân các điểm ảnh với hệ số f_x và f_y . (Pixel co-ordinate transformation)
- Tiếp đến, nội suy giá trị màu cho ảnh vừa được scale trên bằng phương pháp Linear interpolation (Brightness interpolation).

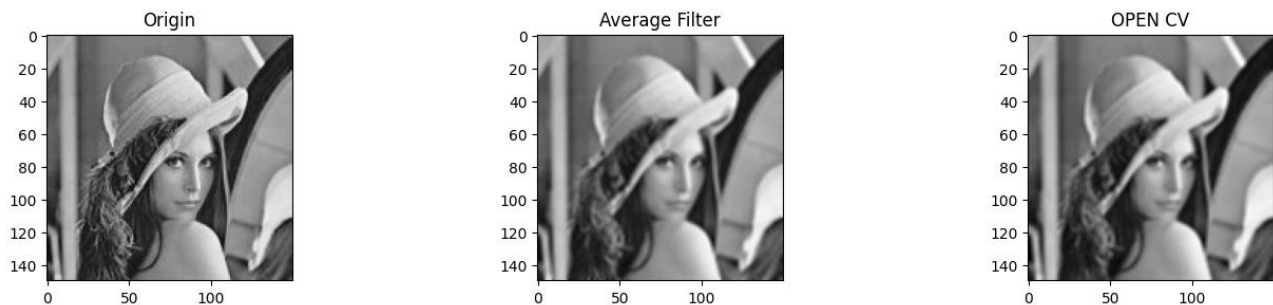
Nhận xét

- Tổng quát
 - o Có thể thấy, dù hệ số f_x và f_y khá lớn, nhưng ảnh A vẫn giữ được các đường nét, thông tin, nội dung của ảnh một cách khá tốt.
 - o Tuy nhiên, nếu nhìn kỹ, có thể thấy màu sắc ở ảnh A có chút nhạt hơn so với ảnh gốc, độ tương phản chưa được tốt như ảnh ban đầu.
- So với thư viện
 - o Nhìn chung, giữa ảnh A (từ thuật toán cài đặt) và ảnh B (từ thư viện) không có sự khác biệt về đường nét, thông tin, nội dung ảnh
 - o Tuy nhiên, so về màu sắc, ảnh B giữ được độ tương phản của ảnh gốc trong khi A bị suy giảm.
 - o Đồng thời, chất lượng ở ảnh B có phần tốt hơn so với A, ảnh ít bị nhòe và mờ hơn.

3. Image smoothing

a. Average filter [3]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm `cv2.blur()` (gọi là B):



3.a.1 - Kích thước bộ lọc là 3×3

Phương pháp

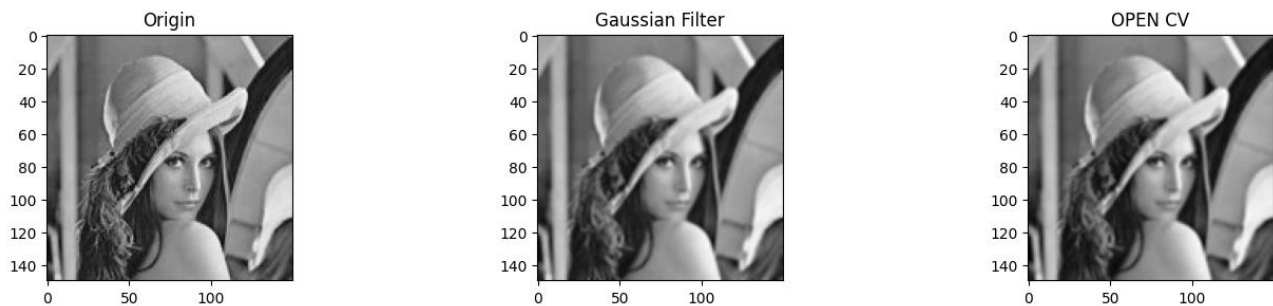
- Sử dụng bộ lọc có kích thước 3×3 để thực hiện thuật toán.
- Duyệt tuần tự các điểm ảnh đầu vào và tính toán với bộ lọc, ta thu được ảnh đầu ra đã được làm mịn.
- Kích thước của bộ lọc (trong trường hợp này là 3×3) có thể được thay đổi để đạt được mức độ làm mịn khác nhau. Bộ lọc lớn hơn có thể tạo ra hiệu ứng làm mịn mạnh hơn, nhưng đồng thời cũng có thể làm mất đi một số chi tiết.

Nhận xét

- Tổng quát
 - Ảnh A đã đáp ứng được yêu cầu đề bài. Các nội dung, chi tiết của ảnh được làm mịn ở mức tương đối (phụ thuộc vào kích thước bộ lọc).
 - Tuy nhiên các biên cạnh cũng đồng thời bị suy giảm và làm mờ.
- So với thư viện
 - Ảnh A và B được sử dụng cùng kích thước của bộ lọc, cả 2 đều được làm mịn.
 - Tuy nhiên, ở B, ảnh được tạo ra với một hiệu ứng làm mờ mềm mại và liên tục. Đồng thời, ảnh ở B vẫn giữ được độ chính xác trong màu sắc cũng như độ tương phản so với ảnh ban đầu.

b. Gaussian filter [3]

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm `cv2.GaussianBlur()` (gọi là B):



3.b.1 - Kích thước kernel là 3 x 3

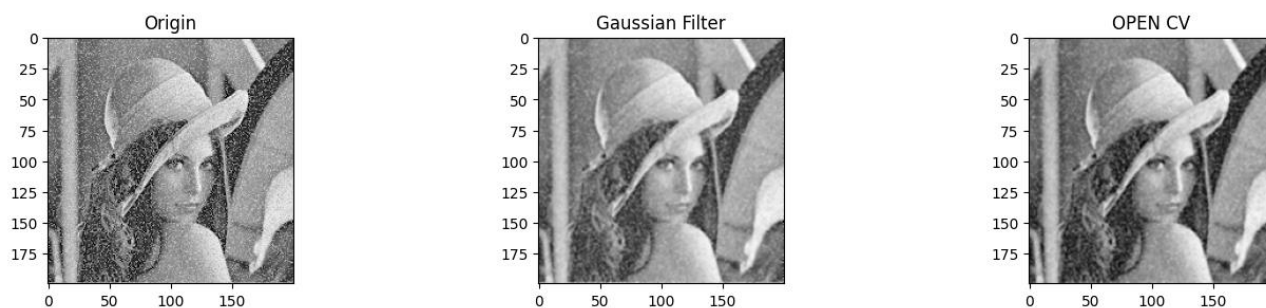
Phương pháp

- Sử dụng kernel có kích thước 3x3 để thực hiện thuật toán.
- Duyệt tuần tự các điểm ảnh đầu vào, thực hiện phép tích chập với kernel, ta thu được ảnh đầu ra đã được làm mịn.
- Kích thước của bộ lọc (trong trường hợp này là 3x3) có thể được thay đổi để đạt được mức độ làm mịn khác nhau. Bộ lọc lớn hơn có thể tạo ra hiệu ứng làm mịn mạnh hơn, nhưng đồng thời cũng có thể làm mất đi một số chi tiết.

Nhận xét

- Tổng quát:
 - Ảnh A đã đáp ứng được yêu cầu đề bài. Các nội dung, chi tiết của ảnh được làm mịn ở mức tương đối (phụ thuộc vào kích thước bộ lọc).
 - Có thể làm tăng độ mịn của ảnh bằng cách thay đổi kích thước bộ lọc lớn hơn và ngược lại cho trường hợp giảm xuống.
 - Các giá trị điểm biên của ảnh không bị suy giảm nhiều.
- So với thư viện:
 - Kết quả ở A và B khá tương đồng nhau: chất lượng làm mờ không có sự khác biệt lớn.

- Tuy nhiên, khi xét các ảnh có độ nhiễu (tại 3.b.2), ảnh B được tạo từ hàm thư viện có khả năng làm mịn tốt hơn, giảm nhiễu hiệu quả hơn so với hàm tự cài đặt.
- Màu sắc mà B đem lại có sự “hài hòa”, giữ được độ tương phản của ảnh gốc.

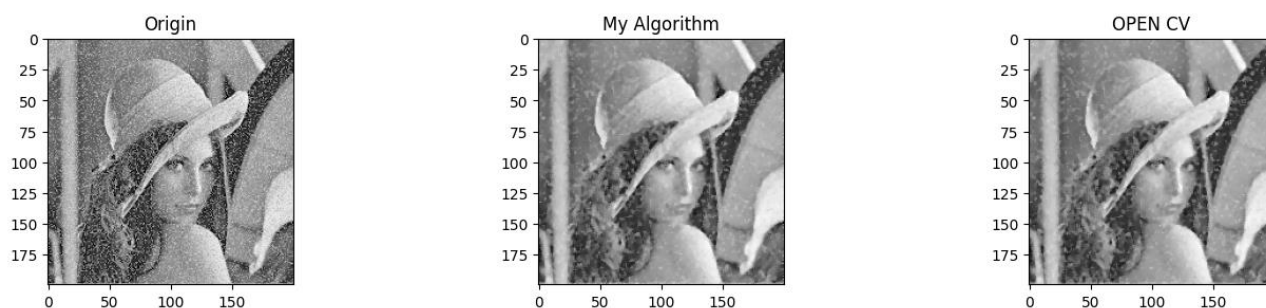


3.b.2 - Áp dụng trên ảnh có độ nhiễu

c. Median filter [3]

Trong phương pháp này, sử dụng ảnh gốc là một ảnh bị nhiễu (nhiều muối tiêu).

Kết quả sau đây bao gồm ảnh gốc, ảnh kết quả từ thuật toán cài đặt (gọi là A), ảnh kết quả từ việc gọi hàm cv2.medianBlur (gọi là B):



3.c.1 - Kích thước bộ lọc là 3 x 3

Phương pháp [4]

- Sử dụng bộ lọc có kích thước 3x3 để thực hiện thuật toán.
- Tại vị trí mỗi điểm ảnh lấy giá trị của các điểm ảnh tương ứng trong vùng 3×3 của ảnh gốc “lấp” vào ma trận lọc.
- Sắp xếp các điểm ảnh vừa được lấy ra theo thứ tự (tăng dần hoặc giảm dần tùy ý).
- Cuối cùng, gán điểm ảnh nằm chính giữa (Trung vị) của dãy giá trị điểm ảnh đã được sắp xếp ở trên cho giá trị điểm ảnh đang xét của ảnh đầu ra output.
- Kích thước của bộ lọc (trong trường hợp này là 3x3) có thể được thay đổi để đạt được mức độ làm mịn khác nhau. Bộ lọc lớn hơn có thể tạo ra hiệu ứng làm mịn mạnh hơn, nhưng đồng thời cũng có thể làm mất đi một số chi tiết.

Nhận xét

- Tổng quát
 - Ảnh được làm mịn và giảm nhiễu đáng kể.

- Các thông tin, nội dung ảnh, đường nét (ở A) được làm rõ hơn. Bối cảnh cũng như nội dung ảnh được cải thiện đáng kể.
- So với thư viện
 - Cả A và B đều có độ làm mịn (khử nhiễu) tương đối bằng nhau.
 - Tuy nhiên, chất lượng, màu sắc ở B được giữ nguyên (so với ảnh gốc) trong khi các giá trị màu ở A lại bị suy giảm.

III. TÀI LIỆU THAM KHẢO

[1] P. G. s. L. Q. Ngọc, “ENG-DIP-LECTURE 3-LQN”.

[2] P. G. s. L. Q. Ngọc, “ENG-DIP-LECTURE 4-LQN”.

[3] P. G. s. L. Q. Ngọc, “ENG-DIP-LECTURE 5-LQN”.

[4] “Lib4U,” [Trực tuyến]. Available: <https://globlib4u.wordpress.com/2013/02/05/bo-loc-trung-vi-median-filter/>.