

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
XỬ LÝ ẢNH SỐ VÀ VIDEO SỐ
ĐỒ ÁN CUỐI KÌ

TĂNG CƯỜNG CHẤT LƯỢNG ẢNH
IMAGE ENHANCEMENT

Tóm tắt

Các kỹ thuật tiền xử lý ảnh và hiệu quả của chúng
đối với các đặc trưng màu và cạnh của ảnh.

PIXEL

1. Bùi Ngọc Bảo Ân
2. La Ngọc Thùy An
3. Nguyễn Phan Mạnh Hùng

Mục Lục

1.	Đặt vấn đề	3
2.	Tóm tắt nội dung.....	3
3.	Động lực.....	3
4.	Hướng tiếp cận.....	3
5.	Nguyên lý chung của các kỹ thuật tiền xử lý.....	4
5.1.	Spatial Domain.....	4
5.2.	Frequency Domain:	4
6.	Tóm tắt lí thuyết và kết quả thực nghiệm	4
6.1.	Các kỹ thuật đã học	4
6.1.1.	Point Operation	4
6.1.2.	Operator Filter.....	9
6.1.3.	FFT	9
6.2.	Các kỹ thuật tìm hiểu thêm.....	13
6.	Tổng kết.....	26
6.1	Khử nhiễu – Làm trơn – Làm mờ (Deblur, Smoothing)	26
6.2	Làm nét – Làm rõ cạnh (Sharpen)	26
7.	Tài liệu tham khảo.....	26
	Link	26

1. Đặt vấn đề

Đưa ra một số kĩ thuật, phương pháp tiền xử lý thông dụng và đánh giá ảnh hưởng của chúng tác động lên đặc trưng màu, cạnh của Object trên ảnh hoặc toàn ảnh.

2. Tóm tắt nội dung

Tiền xử lý ảnh là một trong những bước cần thiết nhằm làm nổi bật các thông tin cơ sở để giải quyết bài toán nào đó. Thông thường, đối với các bài toán phát hiện, nhận dạng, rút trích đối tượng, đặc trưng màu và cạnh của ảnh cũng là một trong những thông tin cơ sở quan trọng. Nhóm sẽ trình bày các đặc điểm quan trọng của các đặc trưng, đưa ra các kĩ thuật tiền xử lý thông dụng, và đánh giá ảnh hưởng của các kĩ thuật này tác động lên hai đặc trưng trên.

3. Động lực

Rút trích đặc trưng là một trong những bước khó nhất của hầu hết các bài toán. Do vậy, tiền xử lý cũng là một bước quan trọng ảnh hưởng kết quả cuối cùng. Khảo sát các phương pháp và kĩ thuật tiền xử lý tác động lên các đặc trưng của ảnh hay đối tượng đóng góp cơ sở tiếp cận cho các vấn đề, biết được mức độ khả thi của giải pháp, giảm nhiều thời gian, công sức trong việc tìm hiểu, định hướng, lựa chọn cách giải quyết phù hợp. Các đặc trưng được sử dụng ở đây là màu và cạnh, bởi chúng được sử dụng trong nhiều phương pháp rút trích, gần gũi với mắt người, ứng dụng nhiều trong thực tế.

4. Hướng tiếp cận

Đặc trưng màu và cạnh đã được nghiên cứu rất nhiều từ lâu, nên các kĩ thuật xử lý trên 2 đặc trưng này rất đa dạng và phong phú. Do vậy sử dụng 2 đặc trưng này để giải bài toán hứa hẹn mức khả thi cao.

Hai đặc trưng của ảnh được dùng phổ biến là màu và cạnh, bởi màu và cạnh rất gần gũi với con người. Ví dụ:

- Các vật thể trong đời sống có xu hướng có 1 màu chủ đạo, từ đó mắt người phân biệt các vật thể với nhau có thể dựa trên màu sắc.
- Cạnh giúp mắt người xác định hình dáng, phạm vi của vật thể.

5. Nguyên lý chung của các kĩ thuật tiền xử lý

5.1. Spatial Domain

Các kĩ thuật thuộc loại này sử dụng giá trị độ xám của ảnh, tác động trực tiếp lên giá trị điểm ảnh. Có nhiều cách phân loại các kĩ thuật này. Cách phân loại của nhóm như sau:

- Point Operation: Các kĩ thuật thuộc loại này biến đổi các điểm ảnh một cách độc lập, tức là giá trị điểm ảnh của điểm này không ảnh hưởng tới điểm ảnh kia. Ví dụ khi sử dụng các biểu thức biến đổi để điều chỉnh độ sáng hay độ tương phản $G = T(F)$, cứ mỗi giá trị input sẽ cho ra một giá trị output tương ứng, chỉ phụ thuộc vào giá trị của điểm ảnh, không phụ thuộc vào các điểm ảnh khác. Một số kĩ thuật cơ bản của loại này:
 - Quantization
 - Adjust Brightness and Contrast: Linear Transform, Non – Linear Transform, Contrast Stretching...
 - Apply Threshold
 - Histogram Equalization
- Operator Filter: Sử dụng các toán tử để lọc lại ảnh, như Average (Mean), Median, Gauss...

5.2. Frequency Domain:

Tìm miền tần số của ảnh, lọc trên giá trị miền tần số, tìm lại ảnh từ miền tần số mới. Các phép lọc được sử dụng: High – Pass Filter (giữ lại tần số cao), Low – Pass Filter (giữ lại tần số thấp).

6. Tóm tắt lí thuyết và kết quả thực nghiệm

6.1. Các kĩ thuật đã học

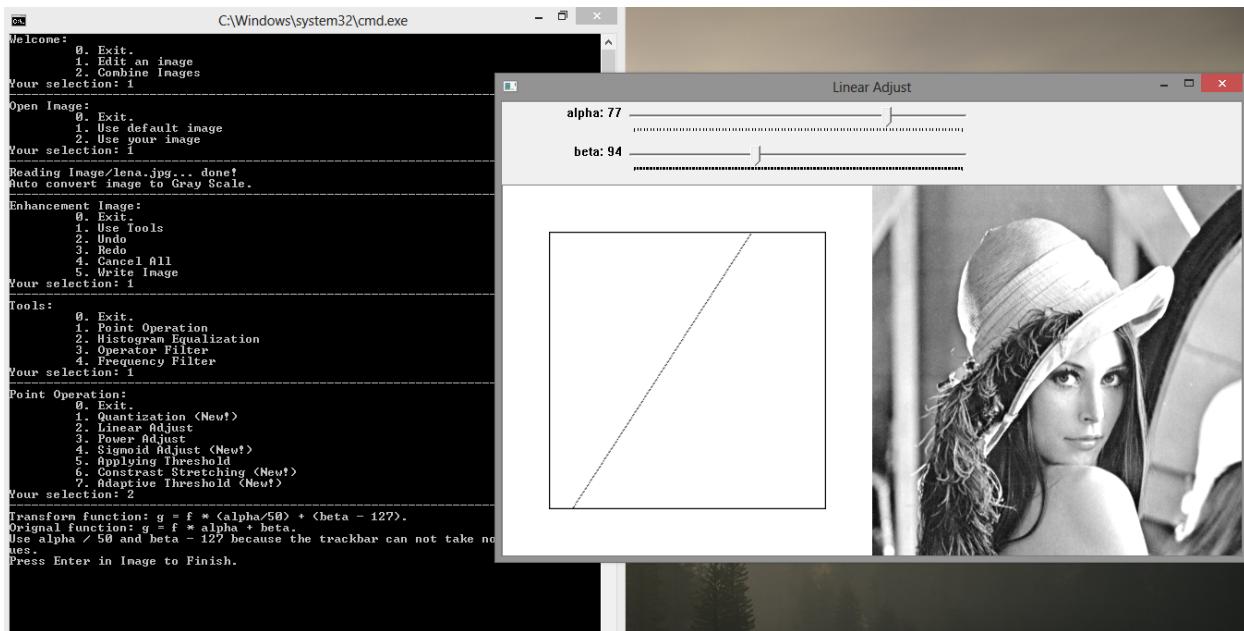
6.1.1. Point Operation

– Điều chỉnh độ sáng hoặc độ tương phản của ảnh bằng các biểu thức biến đổi: $G = T(F)$ với F là ảnh đưa vào, G là ảnh kết quả, T có thể là các hàm:

- Hàm tuyến tính: $T(x) = \alpha * x + \beta$

(Hình minh họa có phương trình là $T(x) = \frac{\alpha}{50} * x + (\beta - 127)$, với α, β là số nguyên không âm.)

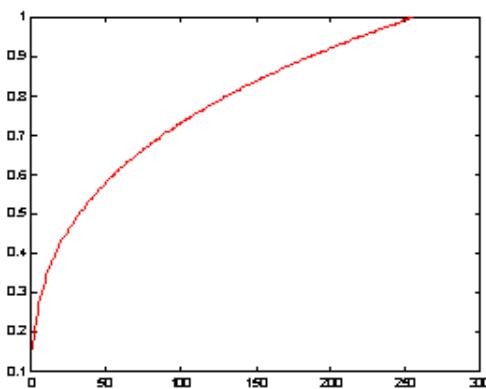
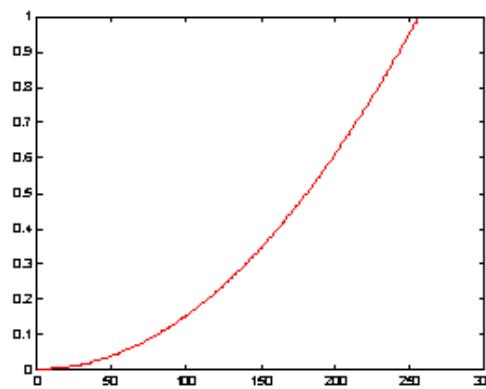
Hàm tuyến tính có hạn chế là dễ dẫn đến bão hòa (giá trị output rơi vào 2 giá trị 0, 255 nhiều), hệ số góc α càng lớn, phương trình càng “thẳng đứng”, độ bão hòa càng cao.



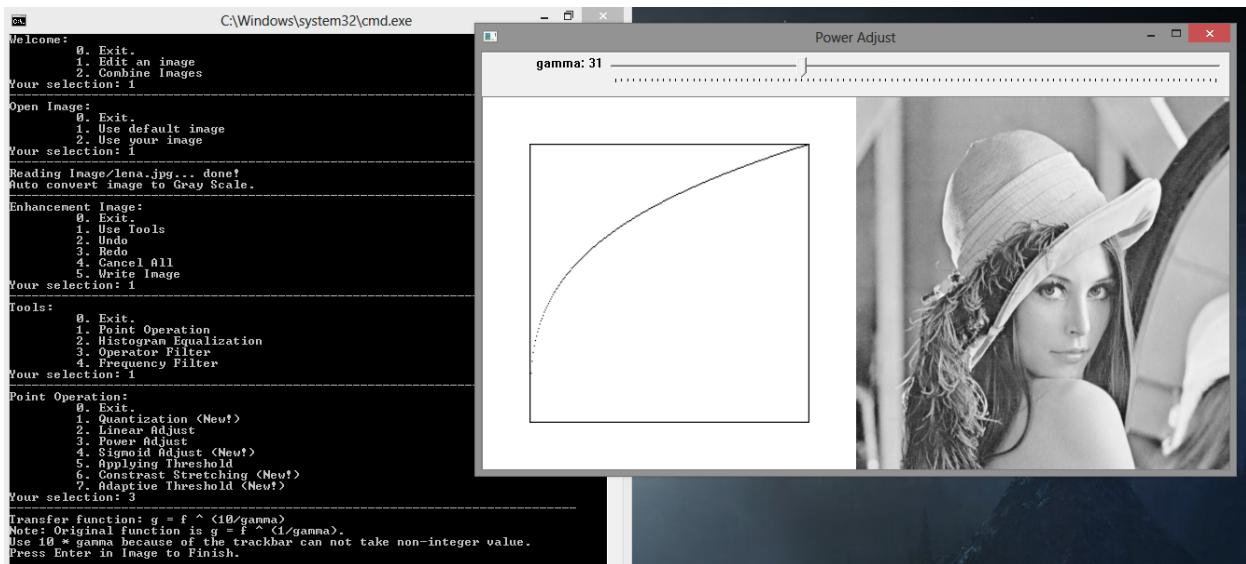
1. Linear Transform

- Hàm phi tuyến: $T(x) = x^{\frac{1}{\gamma}}$

Hàm phi tuyến tránh được nhược điểm của hàm tuyến tính.

1. $\gamma < 1$ 2. $\gamma > 1$

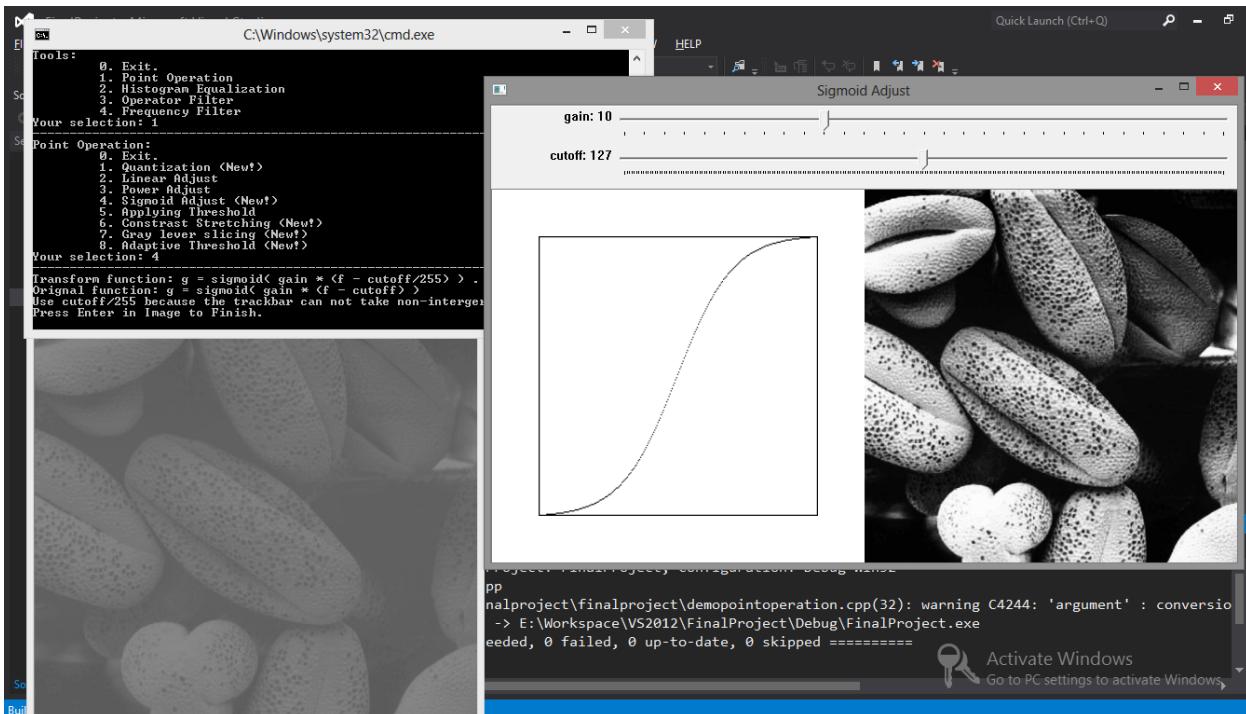
(Hình minh họa có phương trình $T(x) = x^{\frac{10}{\gamma}}$, với γ là số nguyên không âm.)



2. Non-Linear Transform

- Hàm phi tuyén $T(x) = \frac{1}{1+e^{gain * (x - \frac{cutoff}{255})}}$

Với giá trị x quá lớn, e^x sẽ rất lớn, do vậy để sử dụng ta cần đưa ảnh miền $[0, 255]$ về $[0, 1]$. $cutoff$ là giá trị tại đó $T(cutoff) = 0.5$ – tức là tâm của đường sigmoid. Sử dụng miền $[0, 1]$ nên trong minh họa, $T(x) = \frac{1}{1+e^{gain * (x - \frac{cutoff}{255})}}$ với $cutoff$ là số nguyên không âm.



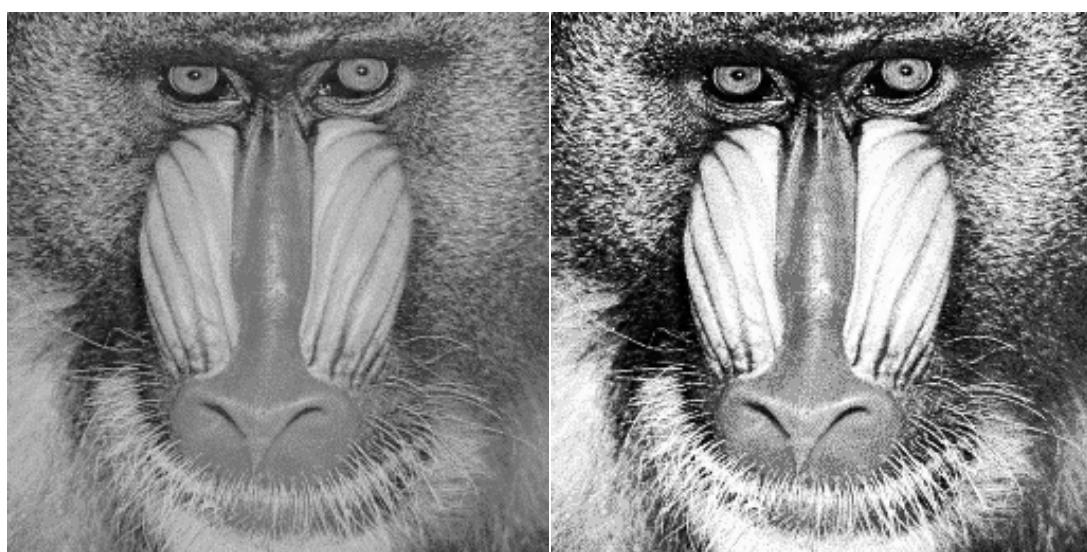
3. Minh họa sử dụng hàm sigmoid

- Đặt ngưỡng: trong ví dụ minh họa, đặt ngưỡng giúp tách object và background. Việc tách background ra khỏi object rất hữu ích cho các bài toán nhận dạng, hoặc để dễ dàng thực hiện các xử lý khác.



3. Threshold = 127

- Cân bằng Histogram



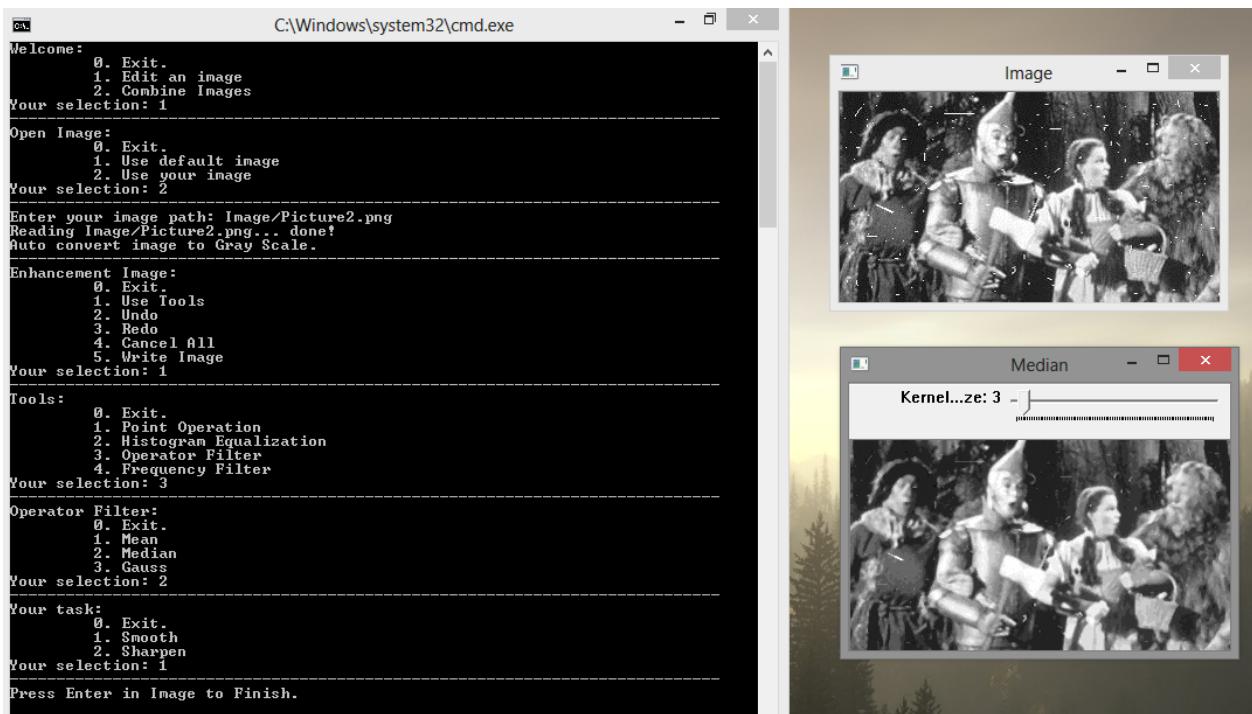
1. Ảnh trước và sau khi cân bằng histogram



2. Ảnh trước và sau khi cân bằng histogram

6.1.2. Operator Filter

Các toán tử được sử dụng: toán tử trung bình, toán tử trung vị, toán tử gauss. Hình minh họa cho thấy sử dụng các toán tử này để làm mờ có thể khử được nhiễu, tuy nhiên làm hình ảnh thiếu rõ nét.



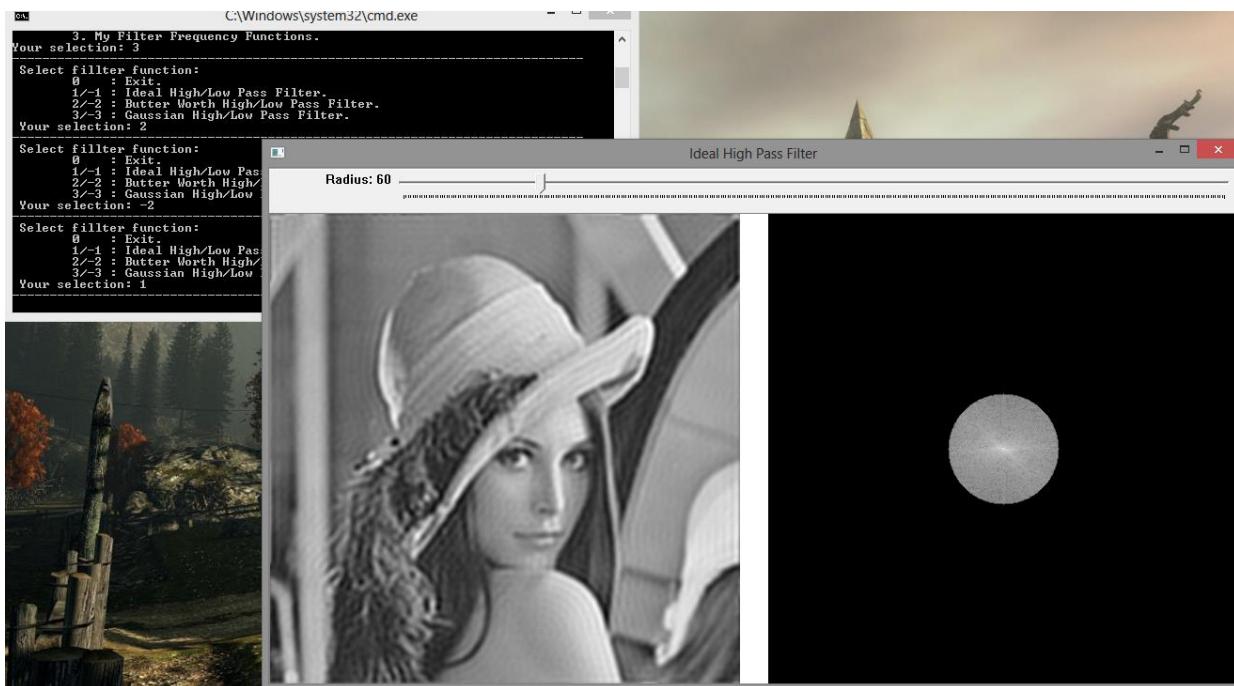
3. Làm mờ bằng toán tử median size 3



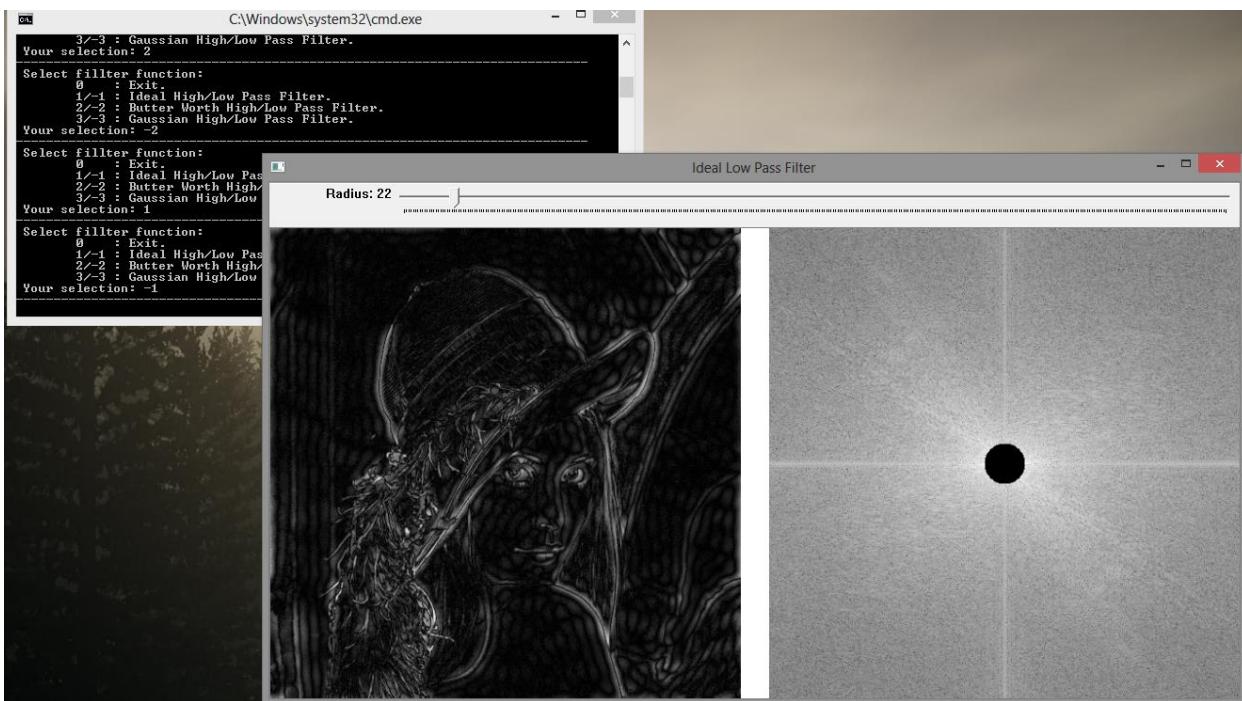
3. Ảnh trước và sau khi làm mờ bằng median, size 3

6.1.3. FFT

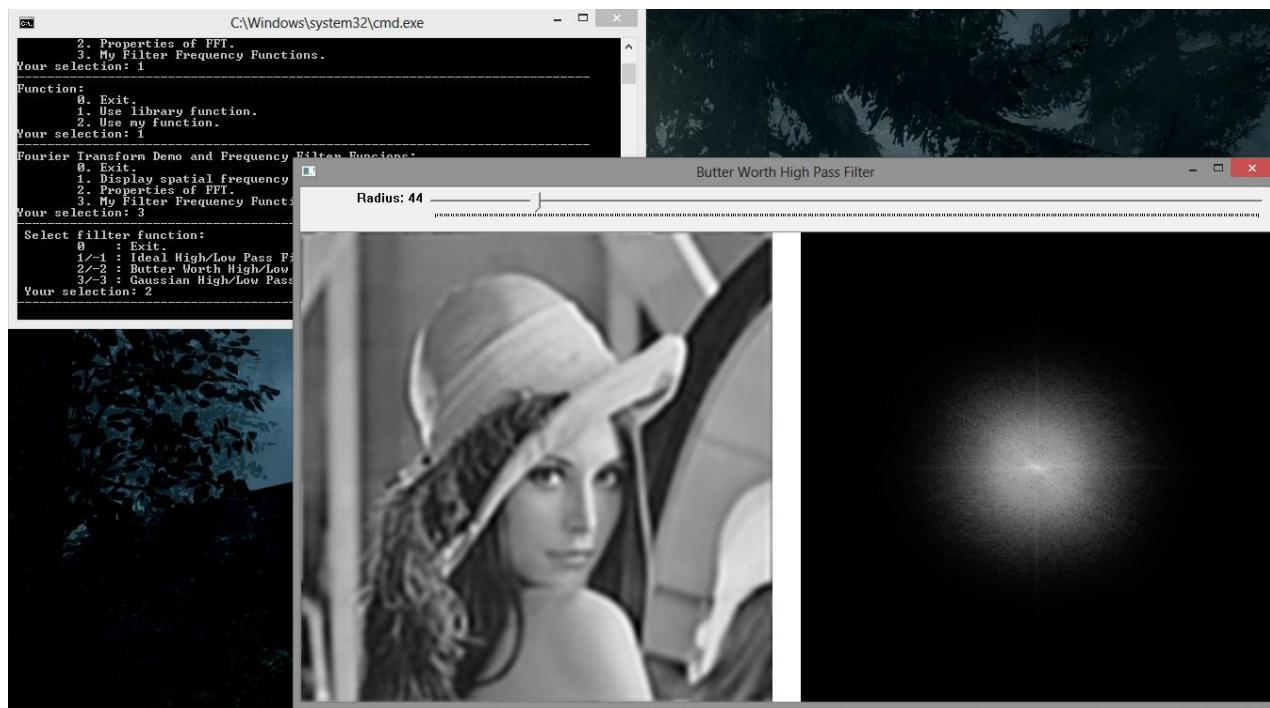
Các phép lọc thông tần số thấp và lọc thông tần số cao sử dụng tương tự bài tập thực hành cuối kì nên không trình bày lại.



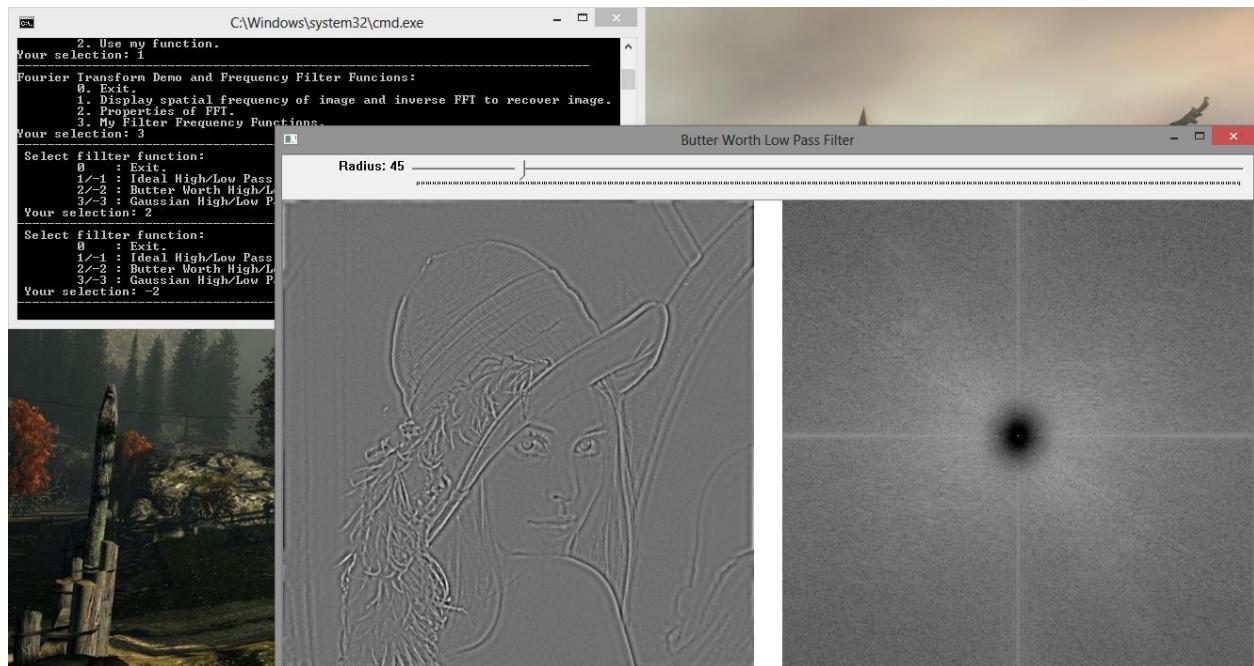
Ideal High Pass Filter



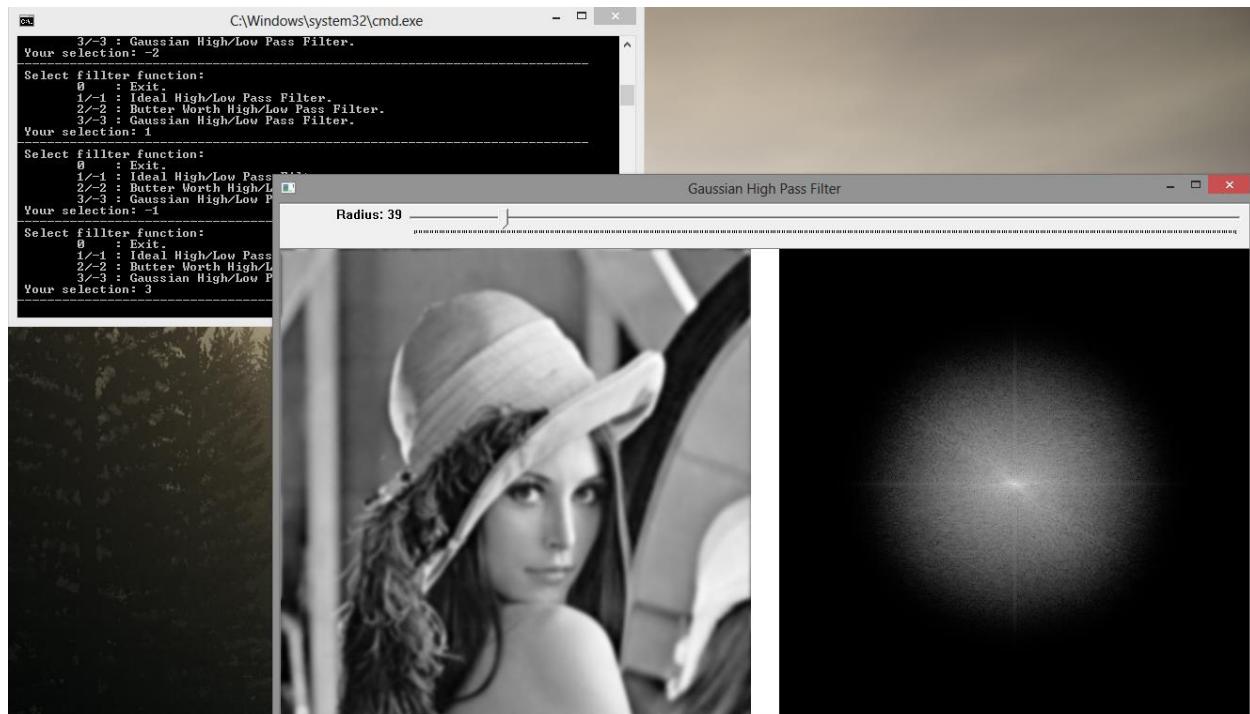
Ideal Low Pass Filter



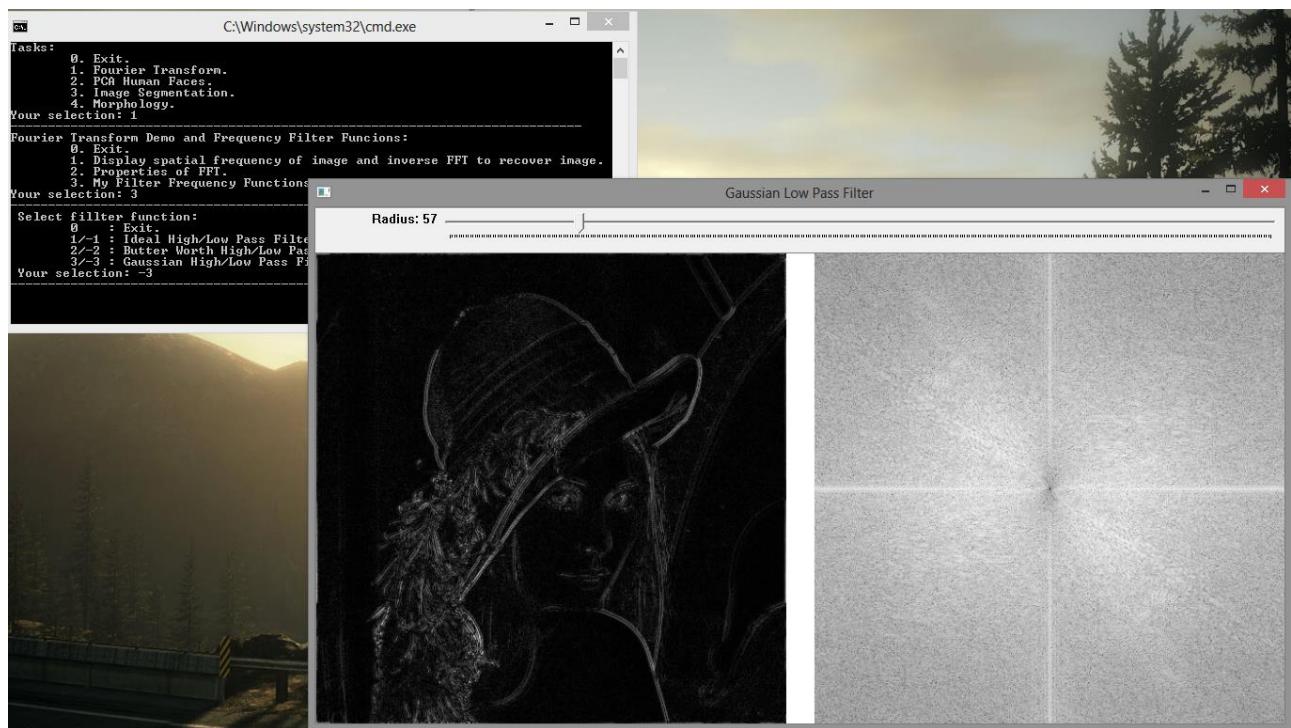
Butter Worth High Pass Filter



Butter Worth Low Pass Filter



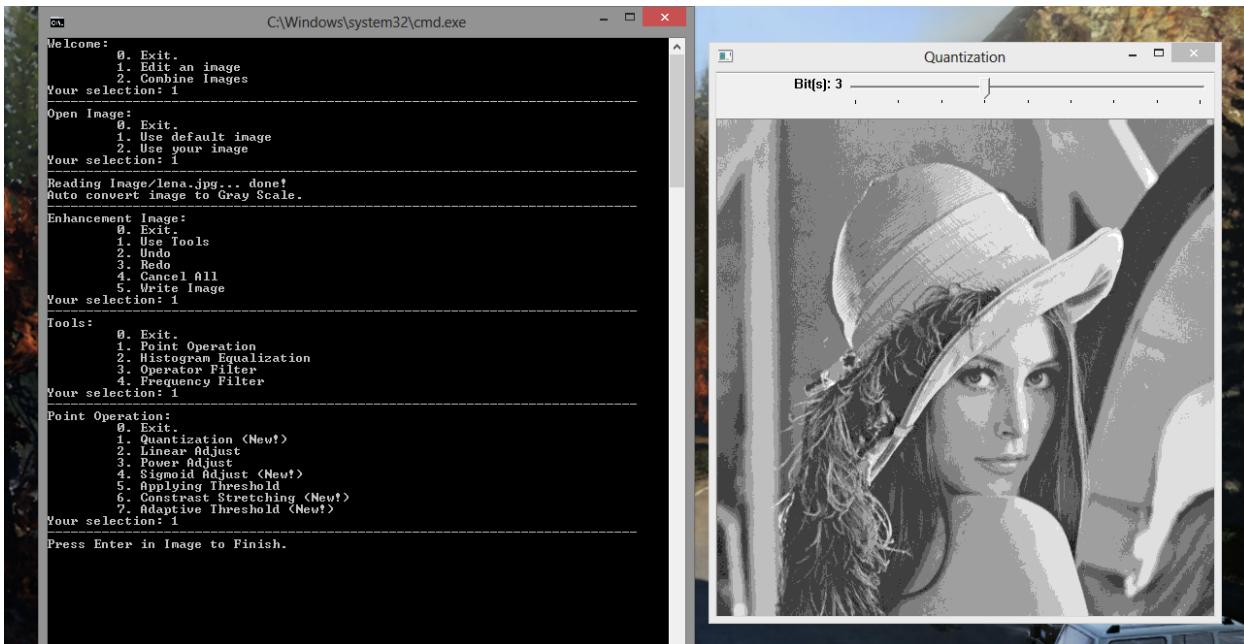
Gaussian High Pass Filter



Gaussian Low Pass Filter

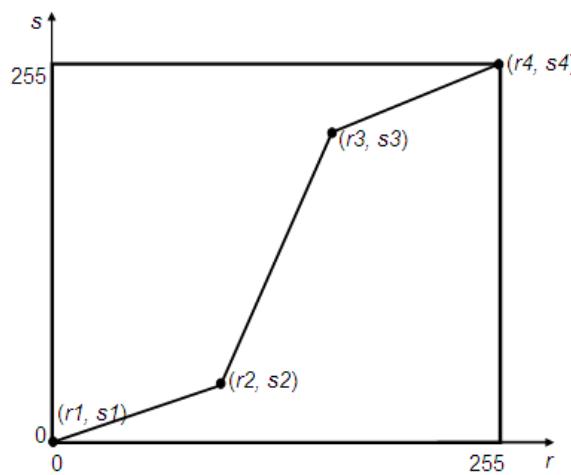
6.2. Các kĩ thuật tìm hiểu thêm

- Quantization: chuyển hóa một ảnh từ lưu ở số bit cao về số bit thấp, giúp ích trong việc lưu trữ ảnh ở dạng nén, tăng tốc độ xử lý cho các kĩ thuật như phân vùng, tìm cạnh, nhận dạng... Tuy nhiên, mắt người nhìn hình này chắc chắn sẽ không đẹp như hình ban đầu.



5. Quantization

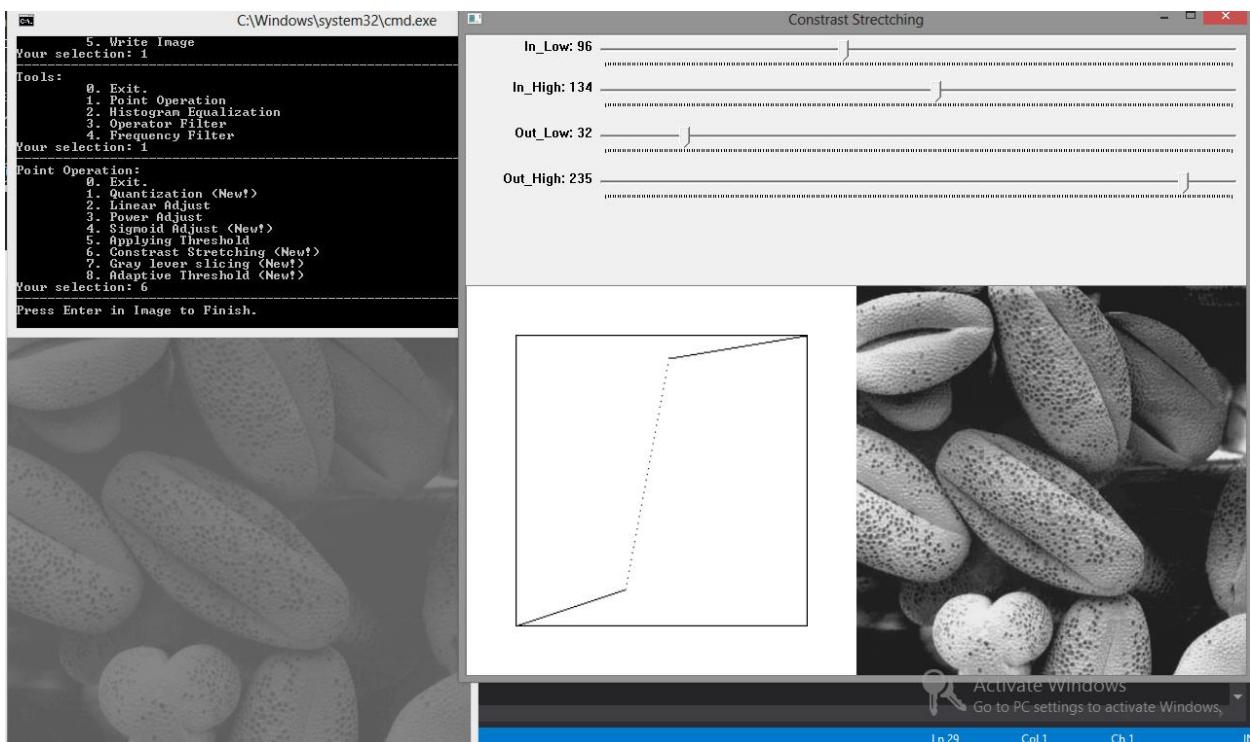
- Contrast Stretching



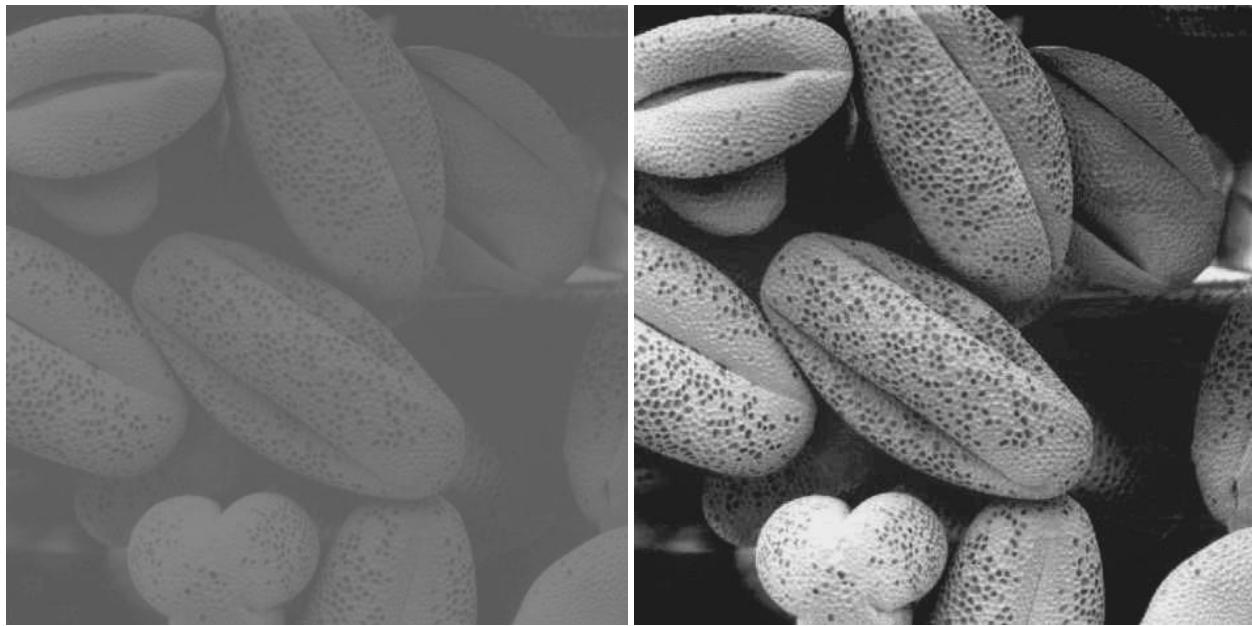
6. Đồ thị contrast stretching

Cặp (r_2, s_2) là cặp (in_low, in_high) , (r_3, s_3) là cặp (out_low, out_high) trong minh họa. Ban đầu khi chưa chỉnh (r_2, s_2) , (r_3, s_3) , đồ thị là đường thẳng từ $(0, 0)$ đến $(255, 255)$

- ảnh output = ảnh input. Sau khi chỉnh (r_2, s_2), (r_3, s_3) ta có hàm chuyển hóa tương ứng. Có thể thấy, khoảng cách giữ r_2, r_3 ban đầu nhỏ, sau khi được điều chỉnh, kéo dãn thành khoảng lớn hơn s_2, s_3 . Do đó chuyển hóa kiểu này được gọi là contrast stretching – làm tăng độ khác biệt, tương phản giữa 2 độ xám. Cũng như mọi kĩ thuật làm tăng độ tương phản khác, cách này giúp ta dễ nhìn hơn, làm rõ cạnh, làm rõ các đối tượng trong ảnh. Nhược điểm của các kĩ thuật làm tăng độ tương phản là có thể làm hình ảnh kém mịn, nhiều mảng màu phân biệt. Tuy nhiên với hình minh họa dưới đây, kết quả khá tốt.

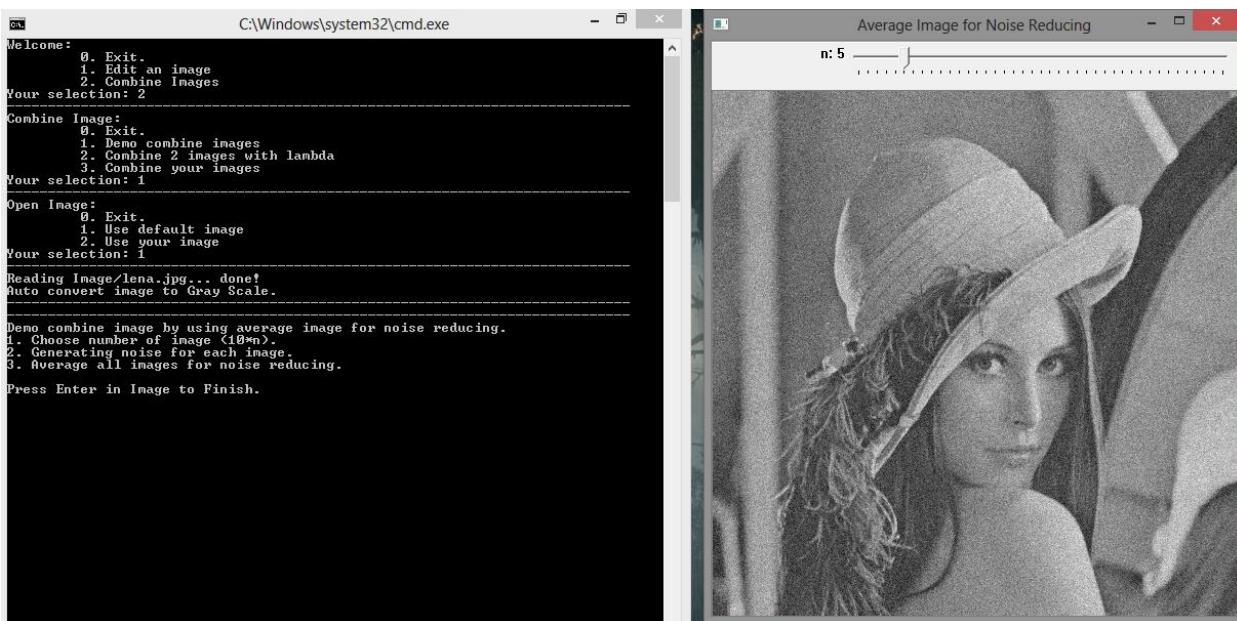


7. Contrast Stretching

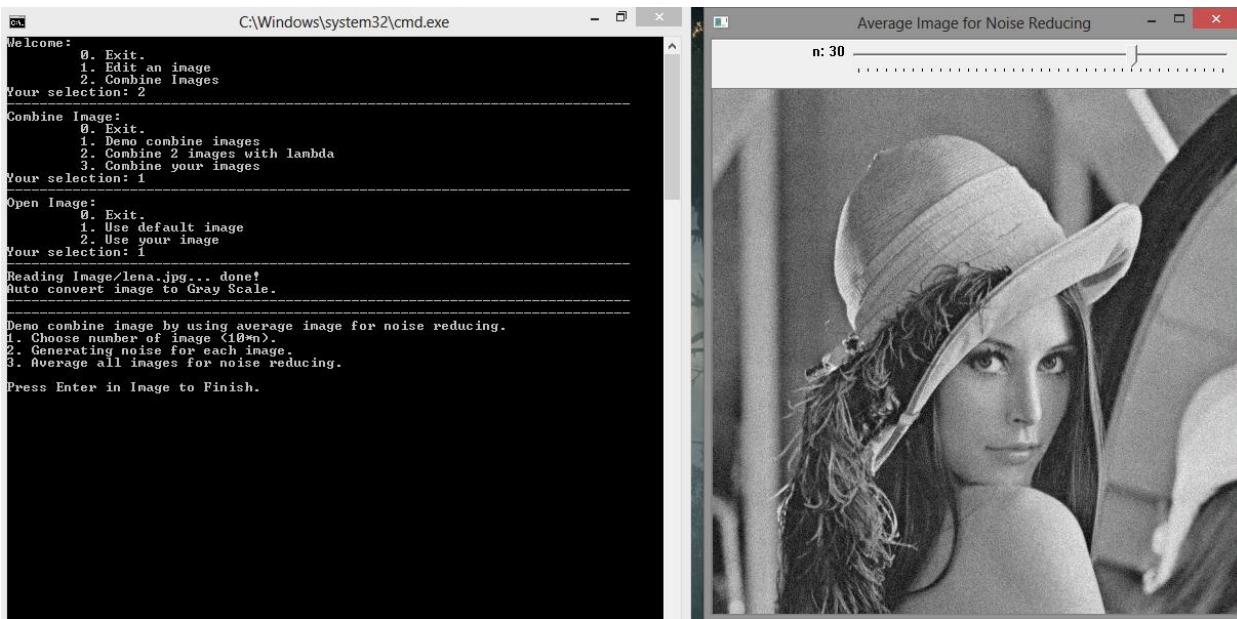


8. Ảnh trước và sau khi áp dụng contrast stretching

- Combine Images (Kết hợp nhiều ảnh)
 - Average Images for Noise Reducing: Lấy trung bình các ảnh nhiễu sẽ giảm được độ nhiễu trong ảnh. Hình bên dưới là thí nghiệm: lấy một hình bất kì, sinh ra n hình, mỗi hình đều được thêm nhiễu ngẫu nhiên. Lấy trung bình các hình, xem kết quả và tương quan của kết quả đó với n. Thí nghiệm cho thấy n càng lớn khả năng giảm nhiễu càng cao.

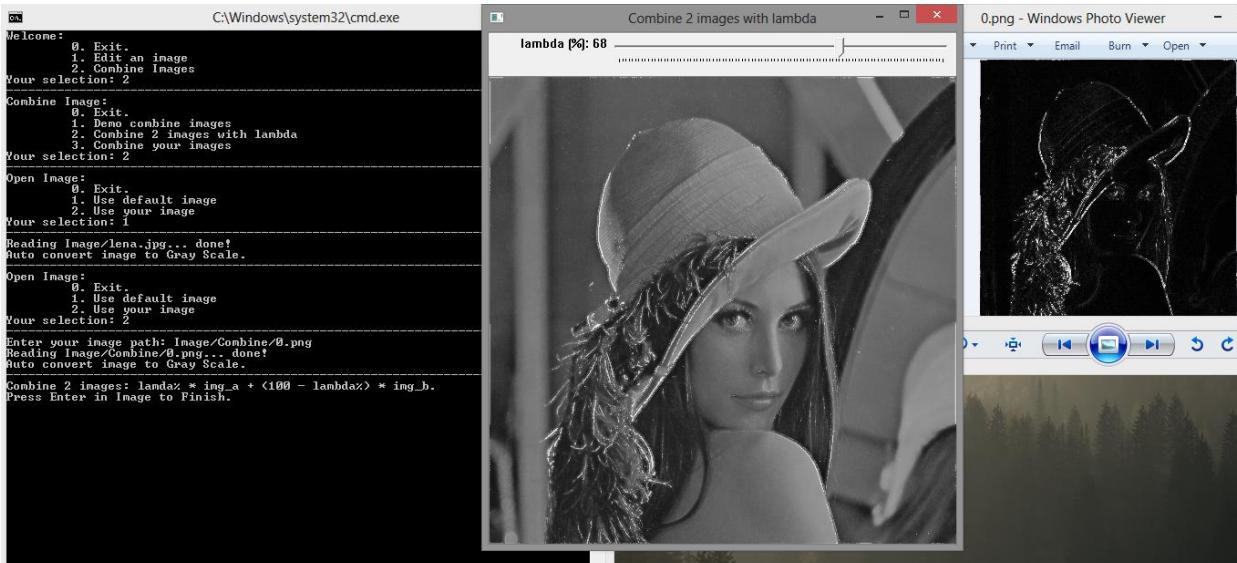


9. Trung bình 5 bức ảnh nhiễu

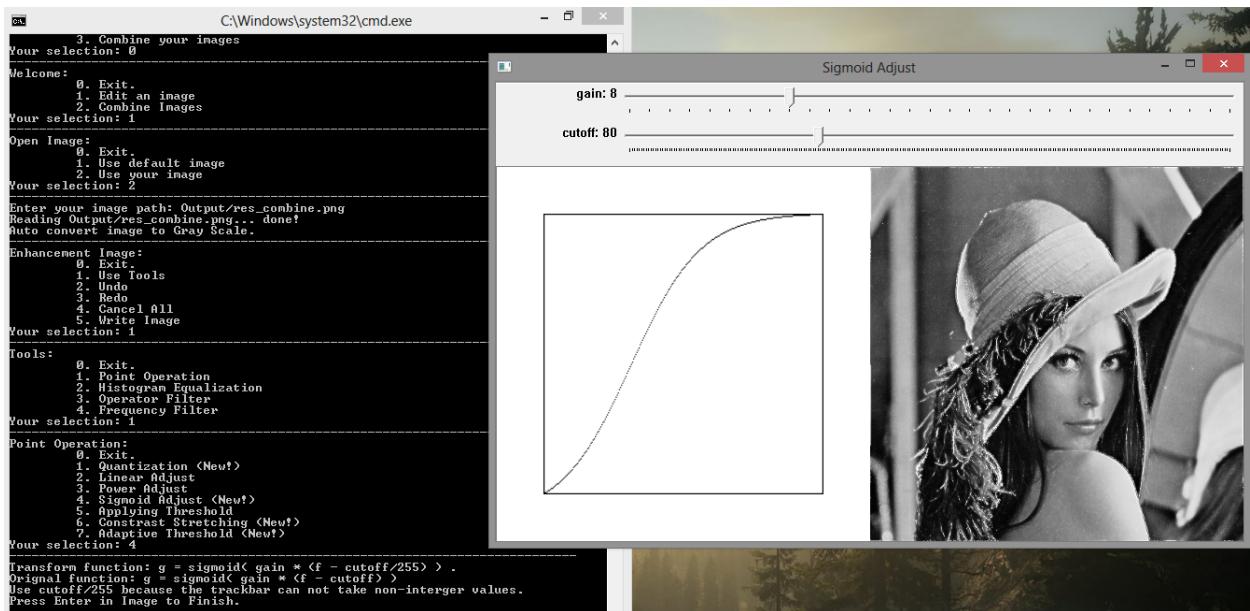


10. Trung bình 30 bức ảnh nhiễu

- Kết hợp 2 hình: hình bên dưới gồm 1 hình gốc và 1 hình kết quả của phép lọc thông tần số High Pass Filter bởi toán tử Gauss kích thước là 7. Khi kết hợp 2 hình lại, ta được hình 11. Hình 11 thấy rõ các cạnh hơn, sắc nét hơn. Do hình được kết hợp với ảnh đã lọc tần số, pha với đa số giá trị màu đen, nên có độ tương phản và độ sáng khá thấp. Để cải thiện kết quả, ta chỉnh lại độ sáng, độ tương phản của hình sau khi kết hợp, được kết quả như hình 12.



11. Kết hợp hình gốc với hình đã được áp dụng lọc thông tần số cao



12. Điều chỉnh lại độ sáng và độ tương phản cho kết quả





13. 2 hình trên: ảnh gốc và ảnh lọc thông tần số cao
2 hình dưới: ảnh sau khi kết hợp và ảnh sau khi chỉnh độ sáng, độ tương phản

Việc kết hợp các ảnh còn có thể ứng dụng trong việc làm mờ ảnh: Thay vì sử dụng trực tiếp kết quả ảnh làm mờ bằng các toán tử trung bình, trung vị, gauss... ta có thể trộn nó với ảnh gốc bằng tỉ lệ nào đó, tăng độ linh động trong việc làm mờ ảnh.

- Sharpen Image
 - Kỹ thuật sử dụng vừa trình bày ở trên cũng là kỹ thuật dùng để làm nét ảnh: kết hợp với ảnh được lọc thông tần số cao và điều chỉnh lại độ sáng, độ tương phản.

Ngoài ra, còn có các kỹ thuật khác:

- Có thể sử dụng tương tự kỹ thuật làm tròn để làm nét ảnh, (tạm đặt là Negative Kernel). Ví dụ với toán tử trung bình với kích thước 3×3 , bảng bên trái là kernel làm tròn ảnh, bảng bên phải là kernel làm mờ ảnh.

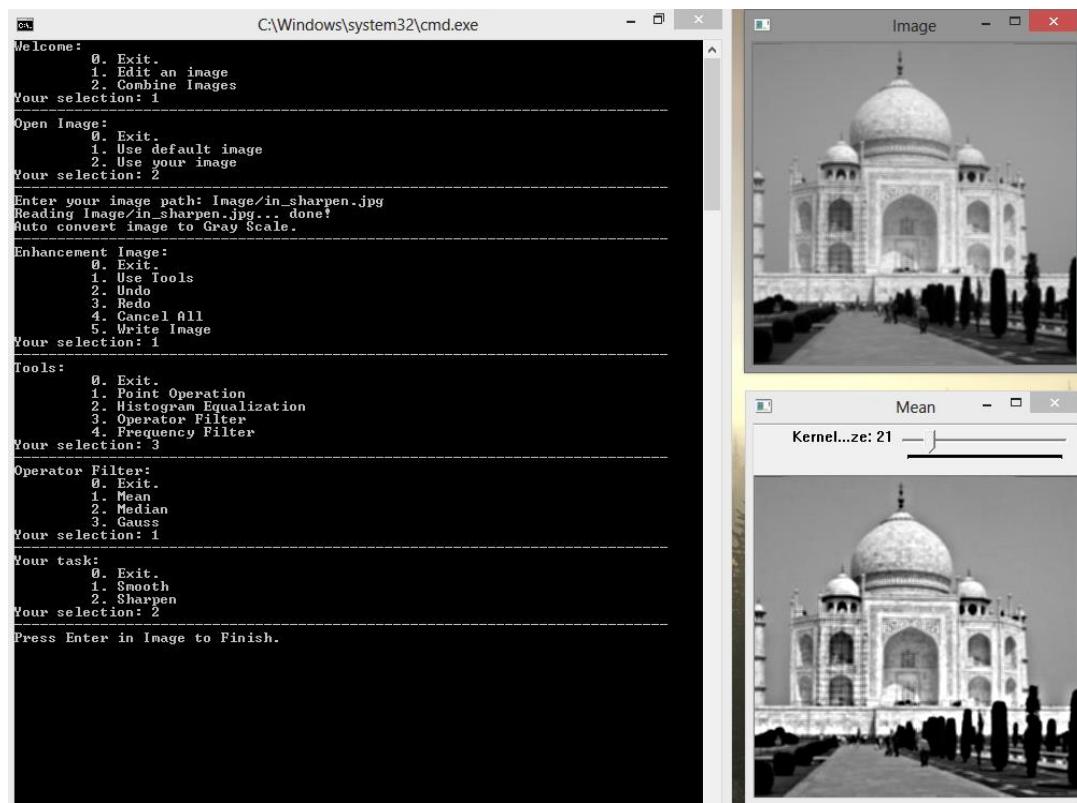
1	1	1
1	-9	1
1	1	1

-1	-1	-1
-1	9	-1
-1	-1	-1

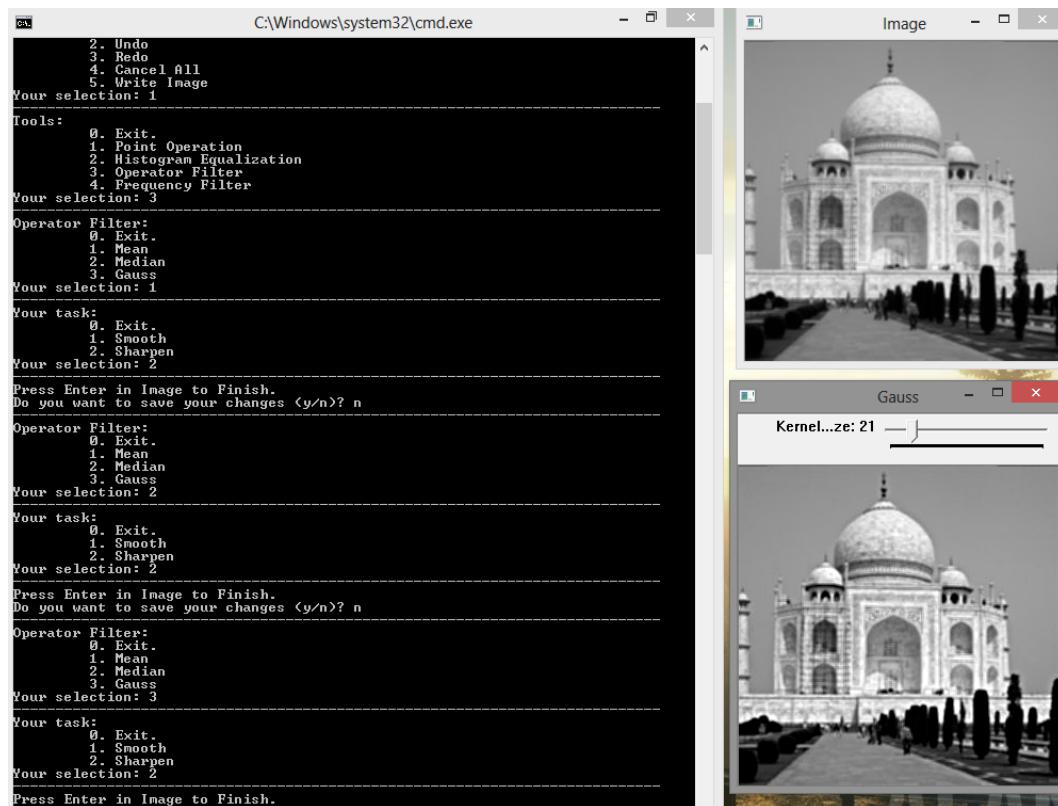
- Sử dụng kết quả của ảnh đã làm mờ, trộn với ảnh gốc theo tỉ lệ:

$$\text{ảnh gốc : ảnh làm mờ} = 1.5 : -0.5.$$

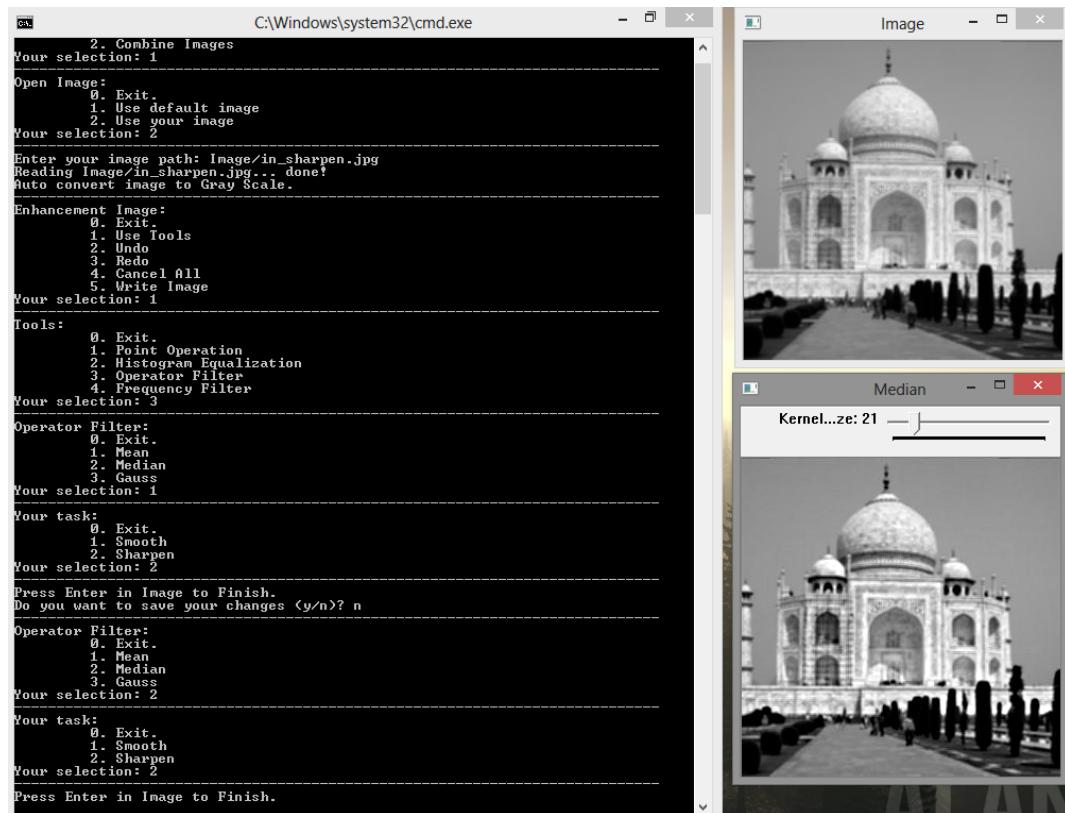
Dưới đây là kết quả sử dụng cách thứ 2: trộn ảnh gốc với ảnh làm mờ để có ảnh nét.



14. Trộn ảnh gốc với ảnh làm mờ sử dụng toán tử trung bình size 21



15. Trộn ảnh gốc với ảnh làm mờ sử dụng toán tử gauss size 21



16. Trộn ảnh gốc với ảnh làm mờ sử dụng toán tử trung vị size 21

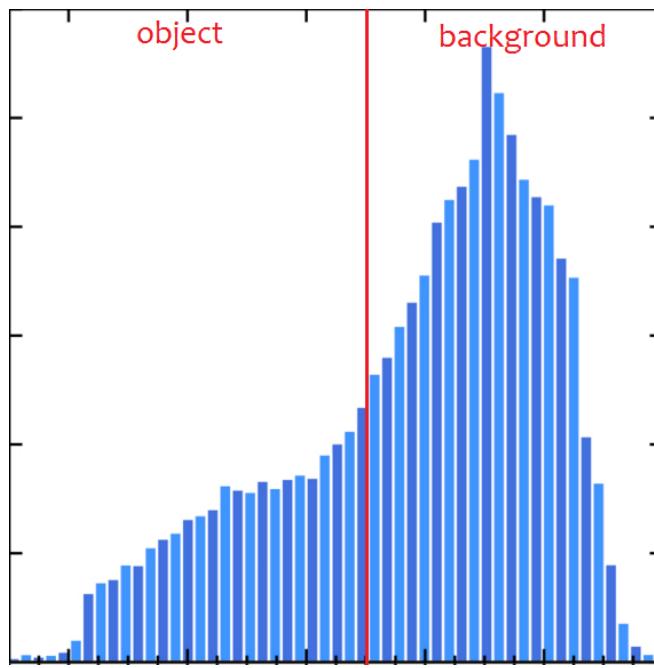


17. Hình trên trái: hình gốc. 3 hình còn lại:
trên phải: dùng toán tử trung bình, dưới trái: toán tử trung vị, dưới phải: toán tử gauss

- Global Thresholding – Otsu's method

Ứng dụng của việc đặt ngưỡng cố định có thể dùng trong việc tách background (nền) khỏi object. Tuy nhiên, thay vì sai và thử để tìm ngưỡng, Otsu đã đưa ra cách tự động chọn ngưỡng khá chính xác.

Nguyên lý:



18. Tìm ngưỡng (đường màu đỏ) và phân vùng cho các giá trị

Dựa vào histogram normalize của ảnh, tính xác suất xảy ra các vùng, từ đó tính phương sai độ khác biệt giữa 2 vùng. Nguồn k chọn được là mức xám sao cho độ khác biệt giữa 2 vùng là lớn nhất.

Cụ thể:

Gọi giá trị histogram normalize của ảnh L mức xám là p

Giả sử ta đã tìm được ngưỡng là k . Khi đó:

- Biết được giá trị nào thuộc vùng background hay object:

$$C_O = \text{pixel} \in [0, k], C_B = \text{pixel} \in [k + 1, L - 1]$$

- Tính $\Pi_O(k)$, $\Pi_B(k)$ là xác suất xảy ra vùng C_O , C_B

$$\Pi_O(k) = \sum_{i=0}^k p(i), \quad \Pi_B(k) = 1 - \Pi_1(k)$$

- Tính được mức xám trung bình của từng vùng theo k : $m_O(k)$, $m_B(k)$

$$m_O(k) = \frac{\sum_{i=0}^k i * p(i)}{\Pi_1(k)}$$

$$m_B(k) = \frac{\sum_{i=k+1}^{L-1} i * p(i)}{\Pi_2(k)}$$

- Tính được giá trị trung bình histogram của dãy $[0, k]$

$$m(k) = \sum_{i=0}^k i * p(i)$$

- Ta có trung bình histogram trên toàn dãy:

$$m_G = \sum_{i=0}^{L-1} i * p(i)$$

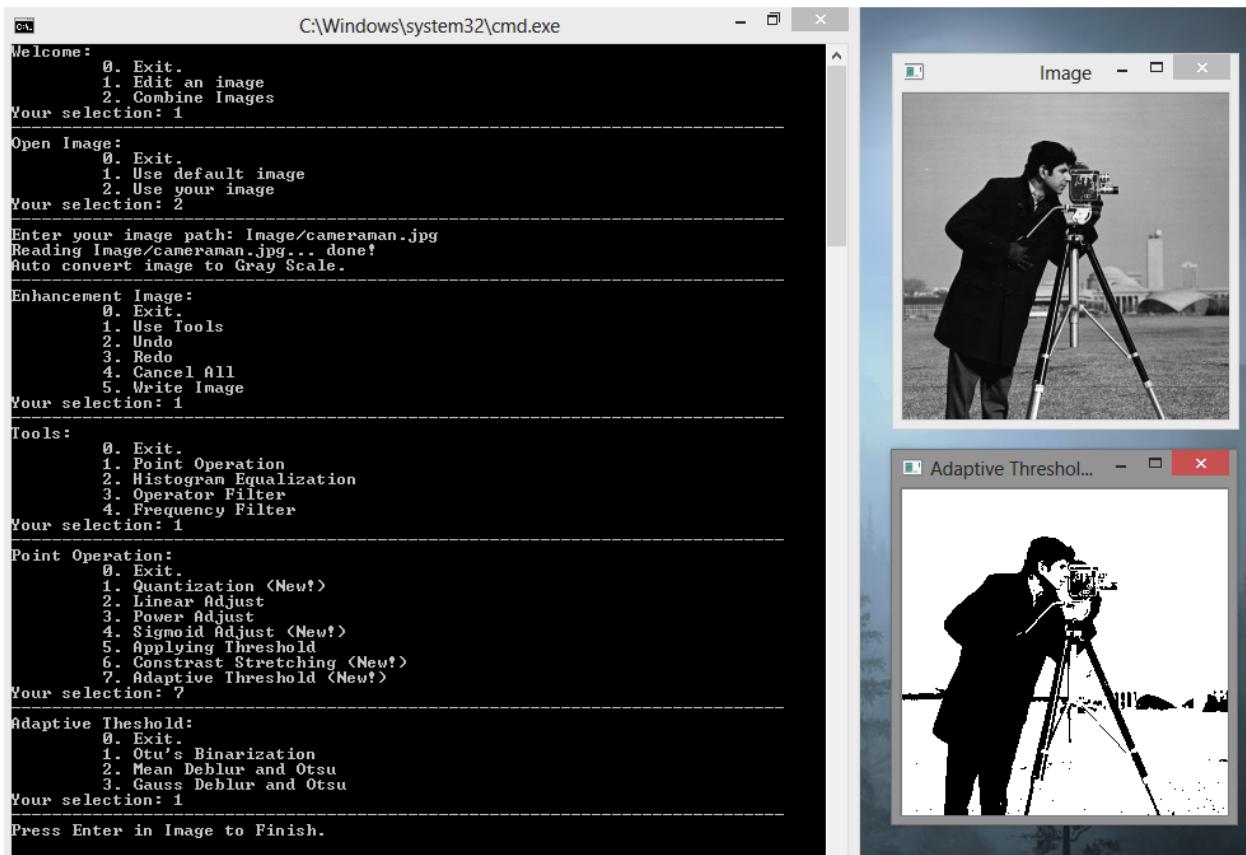
- Độ chênh lệch giữa 2 vùng:

$$\sigma^2(k) = \Pi_O(k) * (m_O(k) - m_G)^2 + \Pi_B(k) * (m_B(k) - m_G)^2$$

Biểu thức trên cũng tương đương với

$$\sigma^2(k) = \Pi_O(k) * (1 - \Pi_O(k)) * (m_1(k) - m_2(k))^2$$

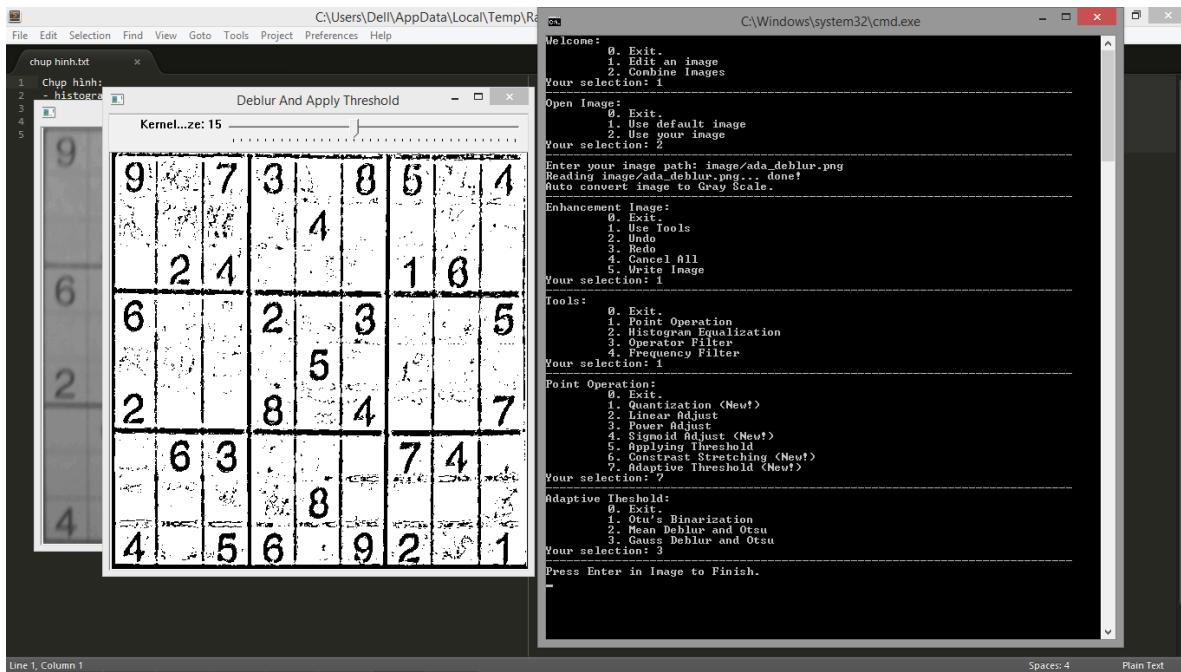
Với mỗi $k \in [0, L - 1]$, ta tính $\sigma^2(k)$ bằng 1 trong 2 biểu thức trên và tìm k sao cho $\sigma^2(k)$ lớn nhất. Một số tài liệu khác tìm 2 giá trị k rồi tính trung bình để làm ngưỡng.



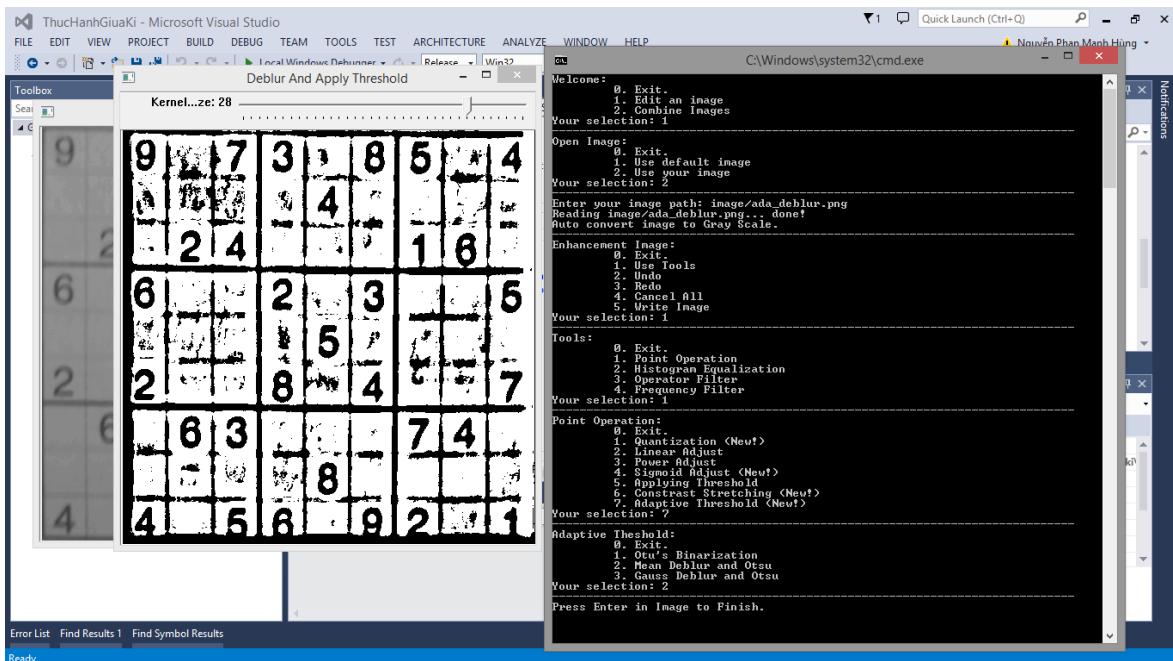
19. Dùng Otsu để tự động tìm ngưỡng

- Adaptive Thresholding

Chương trình được demo dưới đây cũng sử dụng Otsu, nhưng được khử nhiễu bằng một số toán tử trước khi tìm ngưỡng.



20. Dùng Gauss để khử mờ sau đó áp dụng Otsu để tự động tìm ngưỡng

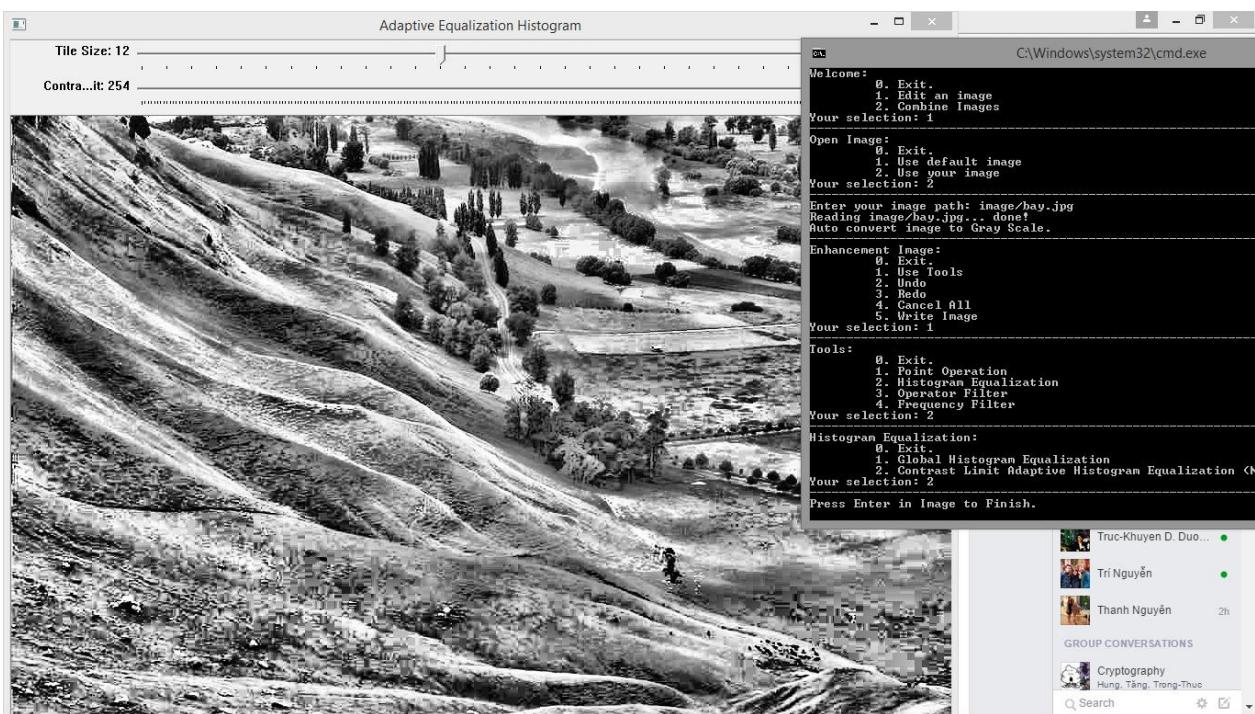
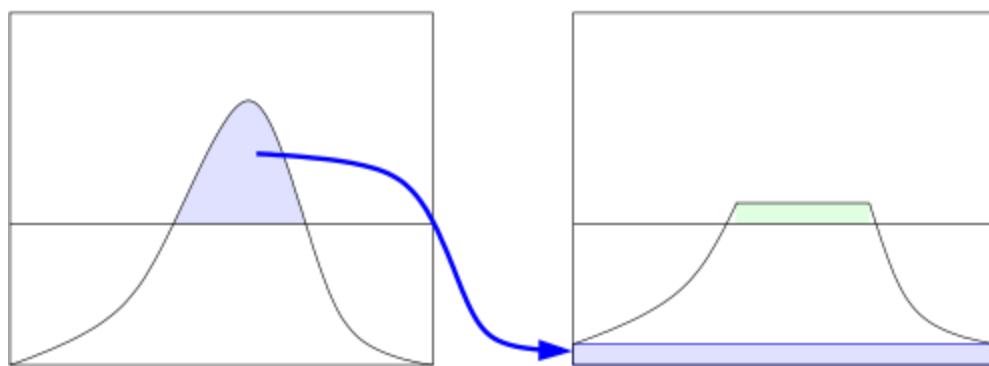


21. Dùng Median để khử mờ sau đó áp dụng Otsu để tự động tìm ngưỡng

- Adaptive Histogram Equalization

Thay vì áp dụng histogram lên toàn ảnh, adaptive histogram áp dụng ảnh lên cân bằng histogram lên từng khung sau đó nối lại. Tuy nhiên do áp dụng lên từng khung riêng lẻ, biên của các khung sẽ hiện ra, nên người ta sử dụng thêm một số loại phân phối (gauss, uniform...) để làm giá trị thay thế, nếu chênh lệch màu giữa 2 biên thấp hơn ngưỡng thì giữ nguyên, nếu lớn hơn ngưỡng thì thay thế giá trị đó.

Contrast limit: Nếu giá trị contrast vượt quá ngưỡng thì san đều cho các giá trị còn lại



22. Adaptive Histogram with Contrast Limit

Áp dụng Adaptive Histogram như trên cải thiện ảnh tốt hơn, tuy nhiên nếu quá lạm dụng, sẽ làm cho ảnh bị nát, bị vỡ. Ngoài ra, đối với các ảnh nhiễu cao, phương pháp này sử dụng không tốt.

6. Tổng kết

6.1 Khử nhiễu – Làm tròn – Làm mờ (Deblur, Smoothing)

- Tăng độ sáng, giảm độ tương phản
- Average Filter, Median Filter, Gauss Filter
- Low-pass Filter
- Average Multi Images

6.2 Làm nét – Làm rõ cạnh (Sharpen)

- Combine Image with High-pass Filter
- Áp dụng các toán tử làm mờ với Negative Kernel
- Trộn với ảnh làm mờ với hệ số âm.
- Tăng độ tương phản, giảm độ sáng
- Histogram Equalization

7. Tài liệu tham khảo

[1] Rafael C. Gonzalez Richard E. Woods - Digital Image Processing - Pearson (2007)

[2] Slide “Image Filtering: Noise Removal, Sharpening, Deblurring”, EE314: Multimedia Communication System - 1, Polytechnic University Brooklyn.

[3] Webdoc “Image Thresholding”, University of Alberta

[4] Washington courses’ materials: “Chapter 5: Filtering and Enhancing Images”.

Link

[2] http://eeweb.poly.edu/~yao/EE3414/image_filtering.pdf

[3] <https://webdocs.cs.ualberta.ca/~nray1/MyWebsite/ImageThresholding.htm>

[4] <https://courses.cs.washington.edu/courses/cse576/book/ch5.pdf>