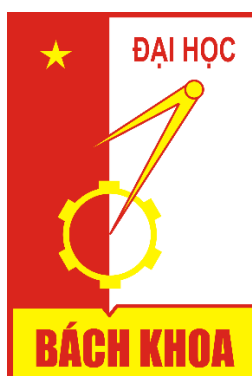


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ VIỄN THÔNG



BÁO CÁO TUYỂN LAB

Đề tài: *Nâng cao chất lượng ảnh bằng cân bằng xám và bộ lọc trung vị*

HOÀNG MINH SƠN- 20193081

son.hm193081@sis.hust.edu.vn

NGÀNH ĐIỆN TỬ VIỄN THÔNG

Người hướng dẫn: Anh Ma Việt Hưng

HÀ NỘI, 05/2020

ĐỀ TÀI BÁO CÁO TUYỂN LAB EDABK

Tìm hiểu lý thuyết về tăng cường chất lượng ảnh bằng điều chỉnh mức xám và bộ lọc làm mịn ảnh Median Filter. Triển khai 2 thuật toán này bằng code C hoặc C++.

Yêu cầu đầu vào đầu ra của chương trình:

- 1) Ảnh đầu vào là ảnh xám kích thước tối đa 512x512 lưu dưới dạng text. Trong đó dòng 1, 2, ..., 512 lưu giá trị độ xám (từ 0 đến 255) các điểm ảnh ở các cột 1, 2, ..., 512 của dòng 1, 2, ..., 512 tương ứng. Các ảnh đầu vào lấy từ bộ dữ liệu ảnh [6]. File text chứa ảnh có thể lấy ở đường link [7].
- 2) Ảnh đầu ra là ảnh sau khi đã được làm mịn hoặc chỉnh độ xám được lưu dưới dạng text sau với định dạng tương tự như ảnh đầu vào.
- 3) Chương trình cần đọc file ảnh txt đầu vào, tiến hành thuật toán cân bằng histogram và/hoặc làm mịn và ghi ảnh đầu ra.

Lời cảm ơn

Trong thời gian làm báo cáo và thực hiện đề tài do lab đề ra, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của anh Ma Việt Hưng – Trưởng nhóm hướng dẫn thực hiện.

Em xin gửi lời cảm ơn chân thành đến các thầy cô của lab đã tạo điều kiện cho em để có thể tham gia thi tuyển vào lab và tạo hướng đi để em có thể cải thiện, phát triển khả năng của bản thân mình.

Em cũng xin chân thành cảm ơn đến các thầy cô giáo đại cương trong trường ĐHBK Hà Nội đã dạy dỗ cho em về mặt kiến thức giúp em có cơ sở và nền tảng để có thể phát triển được khả năng của bản thân trong việc giải quyết các đề tài được giao.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

Hà Nội, ngày.....tháng.....năm.

Sinh viên thực hiện

Sơn

Hoàng Minh Sơn

Tóm tắt nội dung đề tài

Đề tài nghiên cứu về các phương pháp nâng cao chất lượng hình ảnh trong đó nghiên cứu chính về phương pháp nâng cao chất lượng ảnh dùng bộ lọc trung vị và phương pháp cân bằng xám. Được thực hiện bằng ngôn ngữ lập trình C, ứng dụng compiler code online repl.it và ứng dụng MATLAB. Qua đó rút ra các chức năng, lợi ích và ứng dụng của các phương pháp vào các lĩnh vực thực tế trong đời sống. Thực tế kết quả đề tài phù hợp với các vấn đề đã đặt ra, ứng dụng được cho nhiều lĩnh vực trong đời sống đặc biệt trong lĩnh vực nhiếp ảnh và y tế, quân sự. Có thể phát triển đề tài theo nhiều hướng khác nhau và rộng hơn nữa trong lĩnh vực ý tế và an ninh quân sự như ảnh hồng ngoại và chụp xquang chuẩn xác hơn.

Qua việc thực hiện đề tài em cũng học hỏi thêm được nhiều kiến thức hơn Nâng cao khả năng tự học và tự tìm hiểu của bản thân, cải thiện hơn các kĩ năng giải quyết vấn đề...

MỤC LỤC

CHƯƠNG 1. ĐẶT VẤN ĐỀ.....	9
1.1 Khái niệm (Chức năng, nhiệm vụ) của chức năng cân bằng xám và bộ lọc làm mịn ảnh	9
1.1.1 Bộ lọc trung vị	9
1.1.2 Cân bằng xám	14
CHƯƠNG 2. LÝ THUYẾT CƠ BẢN.....	18
2.1 Giới thiệu chung về ảnh, điểm ảnh, ảnh xám và biểu diễn ảnh xám bằng điểm ảnh.	18
2.1.1 Khái niệm chung về ảnh và ảnh xám	18
2.1.2 Ảnh xám và biểu diễn ảnh xám bằng điểm ảnh:	20
2.1.3 Khái niệm histogram, lí do cân bằng histogram	21
2.2 Thuật toán cân bằng histogram	21
2.2.1 Thuật toán.....	21
2.2.2 Lưu đồ thuật toán	24
2.3 Thuật toán làm mịn ảnh bằng lọc trung vị (Median filter)	27
2.3.1 Thuật toán.....	27
2.3.2 Lưu đồ thuật toán	27
CHƯƠNG 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU LƯU TRỮ ẢNH VÀ CÁC THÔNG SỐ LIÊN QUAN ĐẾN HAI THUẬT TOÁN TRÊN	30
3.1 Về chức năng cân bằng xám:	30
3.2 Về phép lọc trung vị.....	30
CHƯƠNG 4. MÃ NGUỒN TRIỂN KHAI HAI THUẬT TOÁN	31
4.1 Thuật toán lọc trung vị.....	31
4.2 Thuật toán cân bằng xám:	33
CHƯƠNG 5. PHƯƠNG ÁN CHẠY THỬ THÍ NGHIỆM.....	35

5.1 Mô tả ảnh đầu vào	35
5.2 Cách thức chạy chương trình	35
5.3 Phương án đánh giá ảnh đầu ra	35
5.4 Kết quả thử nghiệm trên các tệp ảnh đầu vào	35
CHƯƠNG 6. KẾT LUẬN	45
6.1 Kết luận.....	45
TÀI LIỆU THAM KHẢO.....	46
PHỤ LỤC.....	48

MỤC LỤC ẢNH

Hình 1.1 Ảnh trước và sau lọc trung vị	10
Hình 1.2 Camera lọc nhiễu 3 chiều	11
Hình 1.3 Ảnh siêu âm.....	12
Hình 1.4 Ảnh siêu âm.....	12
Hình 1.5 Radar quân sự.....	13
Hình 1.6 Ảnh chụp vệ tinh	13
Hình 1.7 Histogram khi chưa cân bằng.....	14
Hình 1.8 Histogram khi đã được cân bằng.....	15
Hình 1.9 Ảnh chụp x- ray	16
Hình 1.10 Ảnh nhiệt	17
Hình 1.11 Ảnh vệ tinh	17
Hình 2.1 Điểm ảnh	18
Hình 2.2 Ba loại ảnh thường gặp.....	19
Hình 2.3 Ảnh xám và ma trận ảnh biểu diễn ảnh xám	20
Hình 2.4 Histogram	21
Hình 2.5 Ma trận ảnh xám 8×8	22
Hình 2.6 Bảng thống kê Histogram.....	22
Hình 2.7 Bảng thống kê hàm tích lũy.....	23
Hình 2.8 Thuật toán Nhập file vào mảng $a[N][N]$	24
Hình 2.9 Thuật toán lưu mảng thành ma trận txt	25
Hình 2.10 Thuật toán chính cân bằng Histogram.....	26
Hình 2.11 Mô tả thuật toán Median filter.....	27
Hình 2.12 Thuật toán sắp xếp mảng.....	28
Hình 2.13 Thuật toán chính Median Filter	29

Hình 5.1 So sánh ảnh ban đầu và ảnh qua bộ lọc trung vị	37
Hình 5.2 So sánh ảnh ban đầu và ảnh qua bộ lọc cân bằng xám	37
Hình 5.3	38
Hình 5.4	38
Hình 5.5	39
Hình 5.6	39
Hình 5.7	40
Hình 5.8	40
Hình 5.9	41
Hình 5.10	41
Hình 5.11	42
Hình 5.12	42
Hình 5.13	43
Hình 5.14	43
Hình 5.15	44
Hình 5.16	44

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1 Khái niệm (Chức năng, nhiệm vụ) của chức năng cân bằng xám và bộ lọc làm mịn ảnh

Để có một bức ảnh chất lượng tốt, đảm bảo về độ tương phản, sắc nét. Xử lý ảnh là một bước rất quan trọng và không thể thiếu.

Có rất nhiều cách để xử lý ảnh, nâng cao chất lượng ảnh, như tăng độ tương phản, tăng cường mức xám, làm nét ảnh, loại bỏ các loại nhiễu trên ảnh. Như việc sử dụng các bộ lọc tuyến tính hay phi tuyến, cân bằng xám... Trong đề tài này chúng ta xét đến hai phương pháp để tăng cường chất lượng ảnh đó là.

- + Lọc trung vị (median filter)

- + Cân bằng mức xám (histogram equalization)

1.1.1 Bộ lọc trung vị

1.1.1.1 Khái niệm

Lọc trung vị là một kỹ thuật lọc phi tuyến (non – linear) trong miền không gian. Bộ lọc trung vị là một trong những bộ lọc theo thống kê thứ tự. Nó thay thế các giá trị của điểm ảnh bằng trung vị của các mức xám của các điểm lân cận. Nó khá hiệu quả để loại bỏ đối với hai loại nhiễu: nhiễu đốm (speckle noise) và nhiễu muối tiêu (salt-pepper noise). Kỹ thuật này là một bước rất phổ biến trong xử lý ảnh.[1]

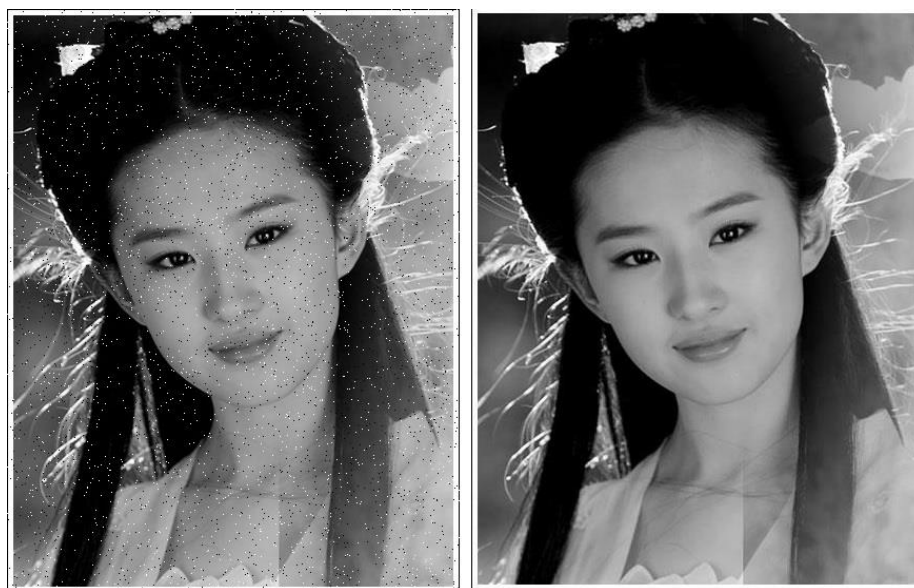
1.1.1.2 Chức năng

Chức năng cơ bản của lọc trung vị là thiết lập giá trị của các điểm với các mức xám khác nhau thành giá trị có vẻ như gần giống với các điểm lân cận.

Qua đó có thể loại bỏ các loại nhiễu và cho ra bức ảnh kết quả sắc nét hơn.

Tuy nhiên việc sử dụng lọc trung vị cũng có điểm hạn chế, chất lượng ảnh đầu ra còn tùy thuộc vào từng loại nhiễu mà ta xử lý. Ví dụ như nhiễu Muối tiêu thì qua lọc trung vị sẽ cho ra chất lượng ảnh được lọc nhiễu và làm trơn tốt vì lọc trung vị sẽ loại bỏ các giá trị cực trị khỏi ảnh input. Phép lọc trung vị giữ lại các

giá trị thật của ảnh input. Còn đối với những bức ảnh bị nhiễu loại khác thì việc sử dụng dạng lọc này có thể đưa ra chất lượng ảnh đầu ra không được tốt [1].



Hình 1.1 Ảnh trước và sau lọc trung vị

1.1.1.3 Ứng dụng thực tiễn

Xét ở khía cạnh nào đó, ta có thể nói **làm mịn ảnh (lọc trung vị)** được ứng dụng khá phổ biến trong nhiều lĩnh vực của đời sống như giải trí, y học, an ninh và một số lĩnh vực khác... Làm mịn ảnh nếu nó đứng riêng lẻ thì hầu như không có ứng dụng gì ngoài công dụng theo nghĩa đen của nó là làm mịn ảnh, và giảm nhiễu. Nhưng khi đặt nó vào trong quy trình xử lý thì rất quan trọng kết quả của nó giúp các xử lý phía sau chính xác hơn. Ví dụ như dò biên với thuật toán canny, trước tiên người ta sẽ loại bỏ nhiễu trước nhằm giúp kết quả dò biên tránh được những sai lầm do nhiễu gây ra. Trong một số trường hợp, các đối tượng thông tin có cùng tính chất với nhiễu (điển hình là trong ảnh siêu âm), việc phát hiện đối tượng khác thường (detect abnormal object) và loại bỏ chúng trước khi tiến hành các xử lý cao hơn là rất quan trọng. Tùy từng đặc thù ảnh và nhiễu mà người ta chọn phương pháp, và sử dụng cửa sổ (design kernel) thích hợp nhằm đạt hiệu quả cao nhất là loại bỏ cái cần loại, giữ cái cần giữ.

Ứng dụng vào công nghệ giám sát “camera lọc nhiễu ba chiều” nhằm tăng cường công tác bảo mật an toàn- an ninh, nếu xảy ra bất cứ một vấn đề hay sự cố gì đều được camera ghi lại, từ đó làm tư liệu, bằng chứng để tìm ra nguyên nhân xảy ra vấn đề.



Hình 1.2 Camera lọc nhiễu 3 chiều

Thêm một ứng dụng có sự góp mặt của làm trơn ảnh là trong công nghệ giám sát rất đáng được nhắc đến là “Ứng dụng công nghệ xử lý ảnh thời gian thực trong bài toán tự động giám sát giao thông tại Việt nam” (được nghiên cứu bởi KS. Lê Quốc Anh, TS. Phan Tương Lai của Trung tâm KHKT & CNQS cùng với PGS. TS. Lê Hùng Lân, ThS. Nguyễn Văn Tiềm của trường ĐH GTVT).

Làm mịn ảnh có ứng dụng rất quan trọng trong y học, chúng ta hãy thử hình dung nếu trong ảnh siêu âm hay chụp nội soi mà xuất hiện những đốm nhiễu, tác động đến phỏng đoán chẩn bệnh của bác sĩ bị sai, dẫn đến phương pháp điều trị sai, hậu quả đe dọa đến sức khỏe, tính mạng con người...quả là vấn đề nghiêm trọng.



Hình 1.3 Ảnh siêu âm



Hình 1.4 Ảnh siêu âm

Chúng ta không thể không nhắc đến ứng dụng của làm mịn ảnh trong lĩnh vực quân sự, một trong những vấn đề quan trọng quyết định sự thịnh suy của một quốc gia. Những hình ảnh thu được từ radar quân sự hay vệ tinh, khi số hóa hoặc gặp sự cố, xuất hiện nhiều làm ảnh hưởng thông tin hữu ích trong ảnh, hay có thể gây hậu quả nghiêm trọng. Tương tự trong lĩnh vực hàng không cho trạm kiểm soát không lưu hay radar trên máy bay



Hình 1.5 Radar quân sự



Hình 1.6 Ảnh chụp vệ tinh

Hiện nay có các phần mềm làm trơn ảnh phổ biến thì phải kể đến Photoshop... Và thú vị hơn cả là “phát triển một số thuật toán xử lý ảnh sử dụng mạng nơ ron tế bào (CNN: cellular Network)” trong đó có thuật toán lọc nhiễu làm mịn ảnh là bộ lọc nhiễu đốm thời gian thực dùng công nghệ nơ ron tế bào, đây là công nghệ có tiềm năng ứng dụng phong phú, và điều quan trọng là có thể thực hiện nhiều mô hình tính toán xử lý thời gian thực phức tạp dùng CNN trên phần cứng hoàn toàn. Đây là công nghệ rất được quan tâm hiện nay, hứa hẹn cho

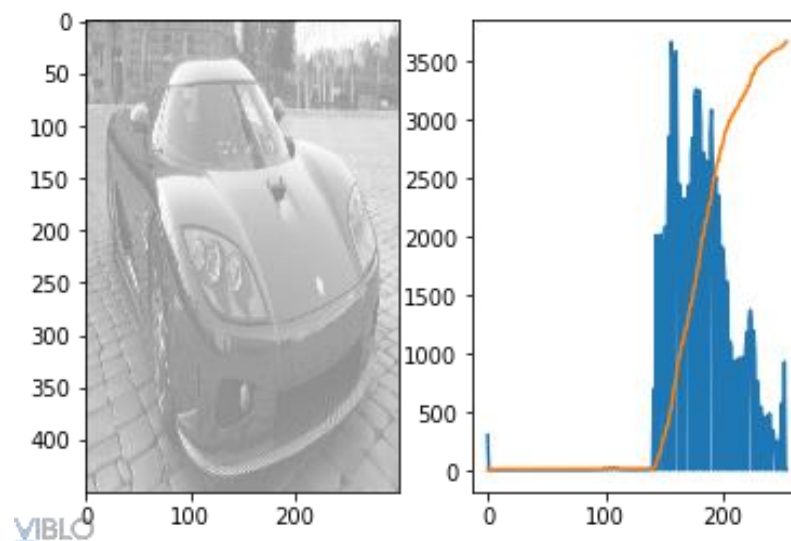
các lĩnh vực nghiên cứu toán học, vật lý, kỹ thuật điện tử về cơ bản và ứng dụng.
[2]

1.1.2 Cân bằng xám

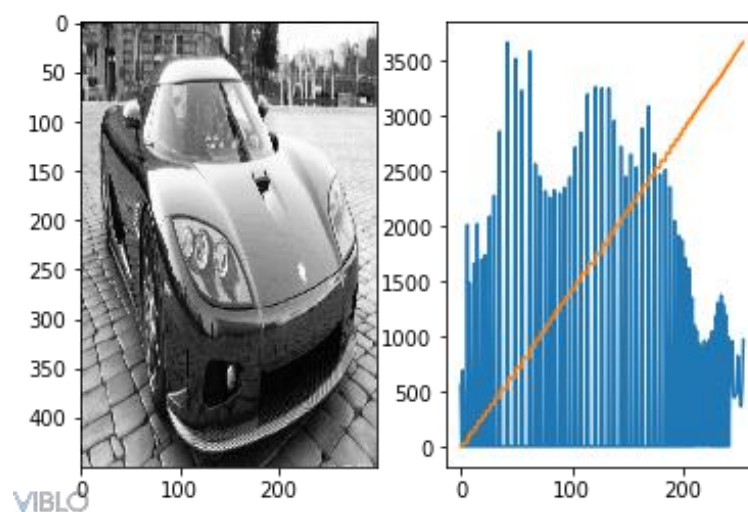
1.1.2.1 Khái niệm

Trước hết ta cần hiểu được khái niệm của histogram, nói một cách đơn giản, ngoài cách nhận biết độ sáng tối của một bức ảnh qua mắt thường, ta có thể nhận biết rõ hơn qua histogram của bức ảnh đó, tôi sẽ thể hiện chi tiết trong chương tiếp theo.

Cân bằng histogram (histogram equalization) là sự điều chỉnh histogram về trạng thái cân bằng, giá trị các điểm ảnh không bị co cụm tại một khoảng nhỏ mà được "kéo dãn" ra. Cân bằng histogram là một phương pháp tiền/hậu xử lý ảnh rất mạnh mẽ. Đặc biệt trong nhiều bài toán trong lĩnh vực computer vision, phương pháp tiền xử lý ảnh này cho chất lượng dữ liệu rất cao, cải thiện chất lượng model deep learning rất nhiều.[3]



Hình 1.7 Histogram khi chưa cân bằng



Hình 1.8 Histogram khi đã được cân bằng

1.1.2.2 Chức năng

Chức năng của cân bằng histogram là giúp cân bằng độ sáng, tối hay làm cho mức xám được phân bố đồng đều để cho ra một bức ảnh với độ tương phản thích hợp nhất.

Tuy nhiên việc sử dụng cân bằng xám cũng phải được thực hiện theo từng trường hợp để tránh các lỗi đáng tiếc như nó có thể làm tăng lên sự tương phản của độ nhiễu cũng như làm giảm đi sự hữu dụng của tín hiệu.[3]

1.1.2.3 Ứng dụng thực tiễn

Cũng giống như bộ lọc làm mịn ảnh, cân bằng xám có rất nhiều ứng dụng quan trọng trong thực tiễn

Trước hết là về lĩnh vực nhiếp ảnh, việc hiểu biết về cân bằng xám sẽ giúp các nhiếp ảnh gia có thể chụp và chỉnh sửa sao cho có thể tạo ra được những bức ảnh với chất lượng tốt nhất làm hài lòng khách hàng và người xem.

Tuy nhiên việc sử dụng cân bằng xám thì thường cho ra những bức ảnh đẹp nhưng với những hiệu ứng khiến cho bức ảnh không được giống thực tế.

Trong lĩnh vực y tế, việc cân bằng xám sẽ giúp cho chất lượng của những bức hình chụp x- quang xương được rõ nét, chất lượng cao hơn giúp cho bác sĩ có thể đưa ra những kết luận chuẩn xác để phục vụ cho việc theo dõi điều trị.



Hình 1.9 Ảnh chụp x- ray

Ứng dụng trong ảnh nhiệt hồng ngoại, đây là một ứng dụng thực tế rất phát triển và giúp con người trong nhiều lĩnh vực.

Tạo ra các camera nhiệt giúp phát hiện được các vật thể trong đêm tối, hay sương mù và ứng dụng thiết thực nhất hiện nay đó là trong việc phòng chống dịch bệnh covid giúp phát hiện được mức nhiệt bất thường của hành khách qua các cảng sân bay để có những động thái và cách xử lý tốt nhất để có thể chuẩn bị phòng chống dịch. Nó là công cụ được sử dụng phổ biến nhất trong lĩnh vực bảo trì dự đoán. Nó được sử dụng rộng rãi bởi những người bảo trì để kiểm tra các lỗi trong thiết bị điện và ứng dụng cơ học để phát hiện các điểm nóng trên bề mặt để tìm thấy bất kỳ loại bất thường nào.



Hình 1.10 Ảnh nhiệt

Ứng dụng trong việc chụp ảnh từ vệ tinh. Trong ngành khoa học thiên văn, vũ trụ giúp các vệ tinh ngoài trái đất đưa ra những hình ảnh chụp được về theo dõi trái đất hay các thiên thể, phục vụ quá trình nghiên cứu khoa học vũ trụ cũng như các lĩnh vực về nghiên cứu trái đất.....[3]



Hình 1.11 Ảnh vệ tinh

CHƯƠNG 2. LÝ THUYẾT CƠ BẢN

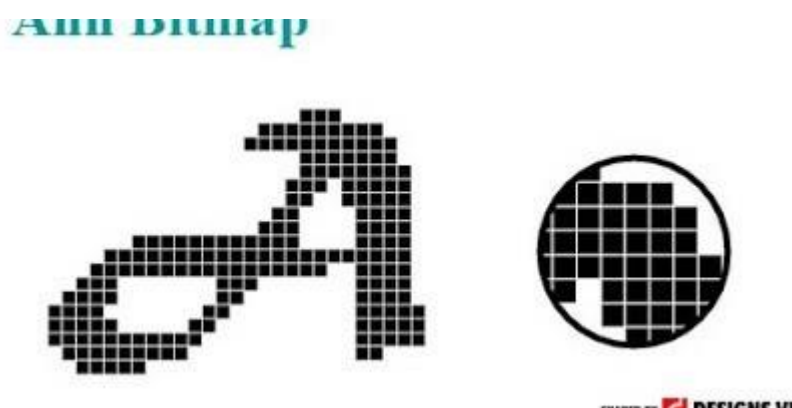
2.1 Giới thiệu chung về ảnh, điểm ảnh, ảnh xám và biểu diễn ảnh xám bằng điểm ảnh.

2.1.1 Khái niệm chung về ảnh và ảnh xám

Trước hết ta cần hiểu những khái niệm cơ bản về ảnh sau:

Ảnh số là tập hợp hữu hạn các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật. Số điểm ảnh xác định độ phân giải của ảnh. Ảnh có độ phân giải càng cao thì càng thể hiện rõ nét các đặc điểm của tấm hình càng làm cho tấm ảnh trở nên thực và sắc nét hơn.

Điểm ảnh (Pixel) là một phần tử của ảnh số tại tọa độ (x, y) với độ xám hoặc màu nhất định. Kích thước và khoảng cách giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh.



Hình 2.1 Điểm ảnh

Các màu sắc cơ sở dùng để biểu diễn một ảnh màu tạo thành một hệ màu và được đề xuất dựa trên các nghiên cứu nhận thức của thị giác của con người. Hệ màu RGB là một trong những mô hình phổ biến nhất trong đó mỗi điểm ảnh được biểu diễn bởi 3 giá trị số thuộc $[0; 255]$.

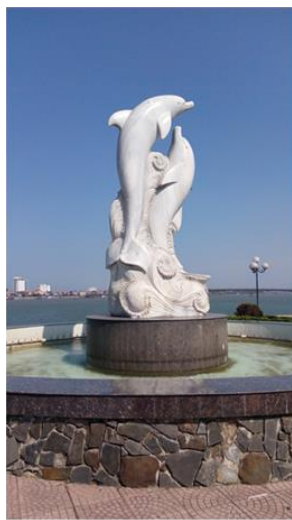
Mức xám: Là kết quả của sự biến đổi tương ứng 1 giá trị độ sáng của 1 điểm ảnh với 1 giá trị nguyên dương. Thông thường nó xác định trong $[0, 255]$ tùy thuộc vào giá trị mà mỗi điểm ảnh được biểu diễn. Các thang giá trị mức xám thông thường: 16, 32, 64, 128, 256 (Mức 256 là mức phổ dụng. Lý do: từ kỹ

thuật máy tính dùng 1 byte (8 bit) để biểu diễn mức xám. Mức xám dùng 1 byte biểu diễn:

$2^8 = 256$ mức, tức là từ 0 đến 255).

Độ sâu màu của ảnh: là đại lượng mô tả khả năng biểu diễn các màu sắc trong ảnh số khi hiển thị trên các thiết bị màn hình hoặc in ấn. Đại lượng này được tính bằng số các bit cần để biểu diễn cho một điểm ảnh

Có thể phân biệt ba loại ảnh: màu, xám, nhị phân qua độ sâu màu với ảnh nhị phân là 1, ảnh xám là 8, ảnh màu là 24 hoặc 32



Color



Grayscale



Black-and-white

Hình 2.2 Ba loại ảnh thường gặp

Độ phân giải (Resolution) của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị. Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên mật độ phân bố, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

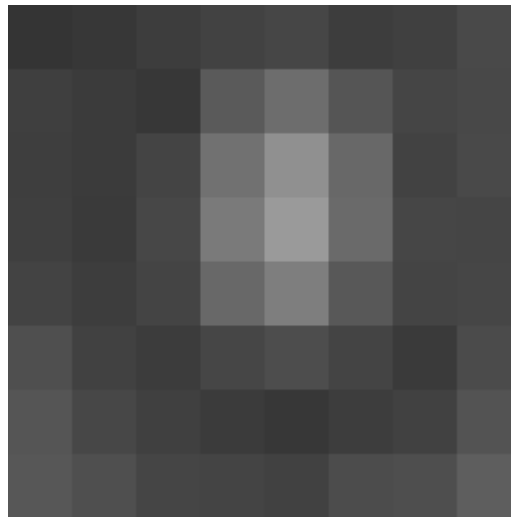
Ví dụ: Độ phân giải của ảnh trên màn hình CGA (Color Graphic Adaptor) là một lưới điểm theo chiều ngang màn hình: 320 điểm chiều dọc * 200 điểm ảnh (320*200). Rõ ràng, cùng màn hình CGA 12" ta nhận thấy mịn hơn màn hình CGA 17" độ phân giải 320*200. Lý do: cùng một mật độ (độ phân giải) nhưng diện tích màn hình rộng hơn thì độ mịn (liên tục của các điểm) kém hơn. [4]

2.1.2 Ảnh xám và biểu diễn ảnh xám bằng điểm ảnh:

Ảnh xám hay còn gọi là ảnh đơn sắc (monochromatic). Ảnh 8 mức xám mỗi điểm ảnh sẽ có giá trị nằm trong đoạn [0-7], ảnh 256 mức xám mỗi điểm ảnh sẽ có giá trị nằm trong đoạn [0-255]. Giá trị của điểm ảnh bằng 0 đại diện cho điểm ảnh tối (đen), giá trị điểm ảnh lớn nhất đại diện cho điểm ảnh sáng (trắng). Độ sáng được tính theo công thức: (chuyển đổi từ hệ màu RGB)

$$S = 0.2989R + 0.5870G + 0.1140B.$$

Sau các định nghĩa ra đưa ra kết luận: Ảnh xám là ảnh mà mỗi điểm ảnh (pixel) trong ma trận ảnh được biểu diễn bởi một số nằm trong [0-255]. Sau đây là ví dụ về biểu diễn ảnh xám:



52	55	61	59	79	61	76	61
62	59	55	104	94	85	59	71
63	65	66	113	144	104	63	72
64	70	70	126	154	109	71	69
67	73	68	106	122	88	68	68
68	79	60	70	77	66	58	75
69	85	64	58	55	61	65	83
70	87	69	68	65	73	78	90

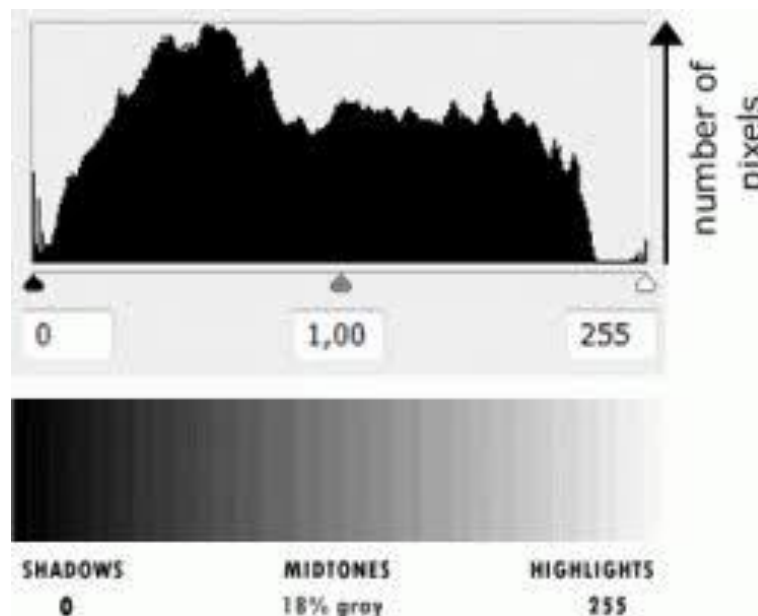
Hình 2.3 Ảnh xám và ma trận ảnh biểu diễn ảnh xám

2.1.3 Khái niệm histogram, lí do cân bằng histogram

Histogram là biểu đồ tần suất thống kê số lần xuất hiện các mức sáng trong ảnh. Histogram của một ảnh số với mức xám thuộc dải xám $[0; L-1]$ là $h(r_k) = n_k$ với r_k là mức xám thứ k , n_k là số điểm ảnh có cùng mức xám thứ k .

Biểu đồ Histogram có trục tung Oy biểu diễn số điểm ảnh của mức xám r_k và trục hoành Ox biểu diễn mức xám r_k

Đối với những bức ảnh tối màu thì biểu đồ tập trung về phía vùng xám thấp (gốc tọa độ), đối những bức ảnh sáng thì biểu đồ tập trung ở phía vùng xám cao và những bức ảnh có độ tương phản thấp thì biểu đồ tập trung ở phía vùng xám giữa. Vì vậy ta cần cân bằng histogram để cho biểu đồ phân bố đều các mức xám. [5]



Hình 2.4 Histogram

2.2 Thuật toán cân bằng histogram

2.2.1 Thuật toán

Để cho thuật toán được rõ ràng ta sẽ xét một ví dụ cụ thể như sau. Xét một ma trận ảnh xám:

52	55	61	59	79	61	76	61
62	59	55	104	94	85	59	71
63	65	66	113	144	104	63	72
64	70	70	126	154	109	71	69
67	73	68	106	122	88	68	68
68	79	60	70	77	66	58	75
69	85	64	58	55	61	65	83
70	87	69	68	65	73	78	90

Hình 2.5 Ma trận ảnh xám 8×8

B1: Thống kê số lượng pixel cho từng mức xám, ta được Histogram H

(i):

Duyệt toàn bộ bức ảnh, tính xem ứng với mỗi bước sáng k có bao nhiêu điểm ảnh và lưu vào mảng **hist[k]** ($k = 0 \rightarrow 255$). Khi đó ta được mảng **hist[k]** có **256 phần tử** lần lượt là các mức xám từ 0-255 và giá trị tương ứng với mỗi phần tử chính là số pixel biểu diễn mức xám đó.

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

Hình 2.6 Bảng thống kê Histogram

B2: Tính "hàm tích lũy" cdf (v) cho từng mức sáng:

Duyệt mảng **hist[k]**, tạo mảng mới **cdfHist[k]** trong đó **cdfHist[0] = hist[0]**, giá trị ban đầu của các phần tử còn lại bằng 0, và **cdfHist[k] = cdfHist[k-1] + hist[k]**.

Sau khi tính toán ta được như hình:

v, Pixel Intensity	cdf(v)	h(v), Equalized v
52	1	0
55	4	12
58	6	20
59	9	32
60	10	36
61	14	53
62	15	57
63	17	65
64	19	73
65	22	85
66	24	93
67	25	97
68	30	117
69	33	130
70	37	146
71	39	154
72	40	158
73	42	166
75	43	170
76	44	174
77	45	178
78	46	182
79	48	190
83	49	194
85	51	202
87	52	206
88	53	210
90	54	215
94	55	219
104	57	227
106	58	231
109	59	235
113	60	239
122	61	243
126	62	247
144	63	251
154	64	255

Hình 2.7 Bảng thống kê hàm tích lũy

B3: Thay thế mức xám với giá trị $h(v)$ tương ứng với $h(v)$ được tính theo công thức:

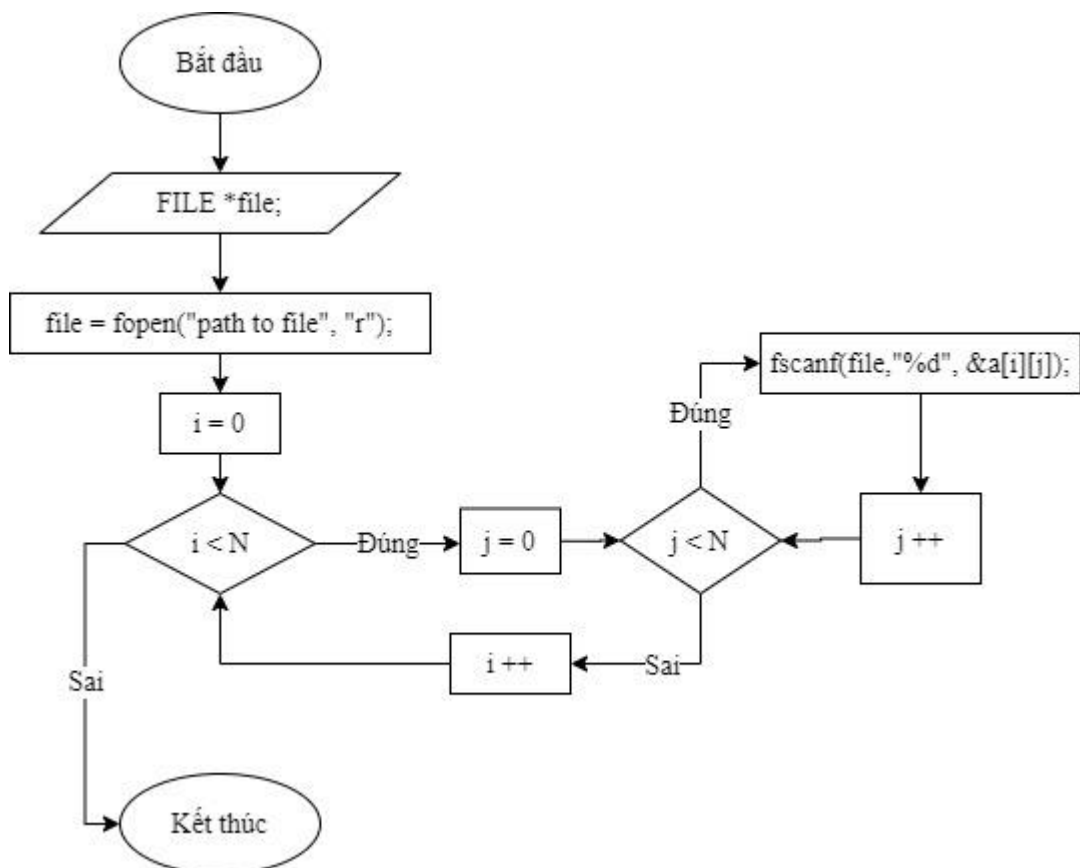
$$h(v) = \text{round} \left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right)$$

Trong đó $M \times N$ là tổng số pixel của ma trận ảnh, giá trị **cdf_{min}** ta xét thấy chính là phần tử **$cdfHist[k]$** tương ứng với phần tử **$hist[k]$** min khác 0. L là số pixel max cùng để biểu diễn ảnh xám, ở đây ta lấy là $L = 255$.

Duyệt mảng ma trận ảnh, với **$a[i][j] = k$** tức là mức xám xuất hiện trong ảnh thì thay bằng giá trị $h(v)$ tương ứng với **$cdfHist[k]$** và lưu vào **$newArray[i][j]$** . [3]

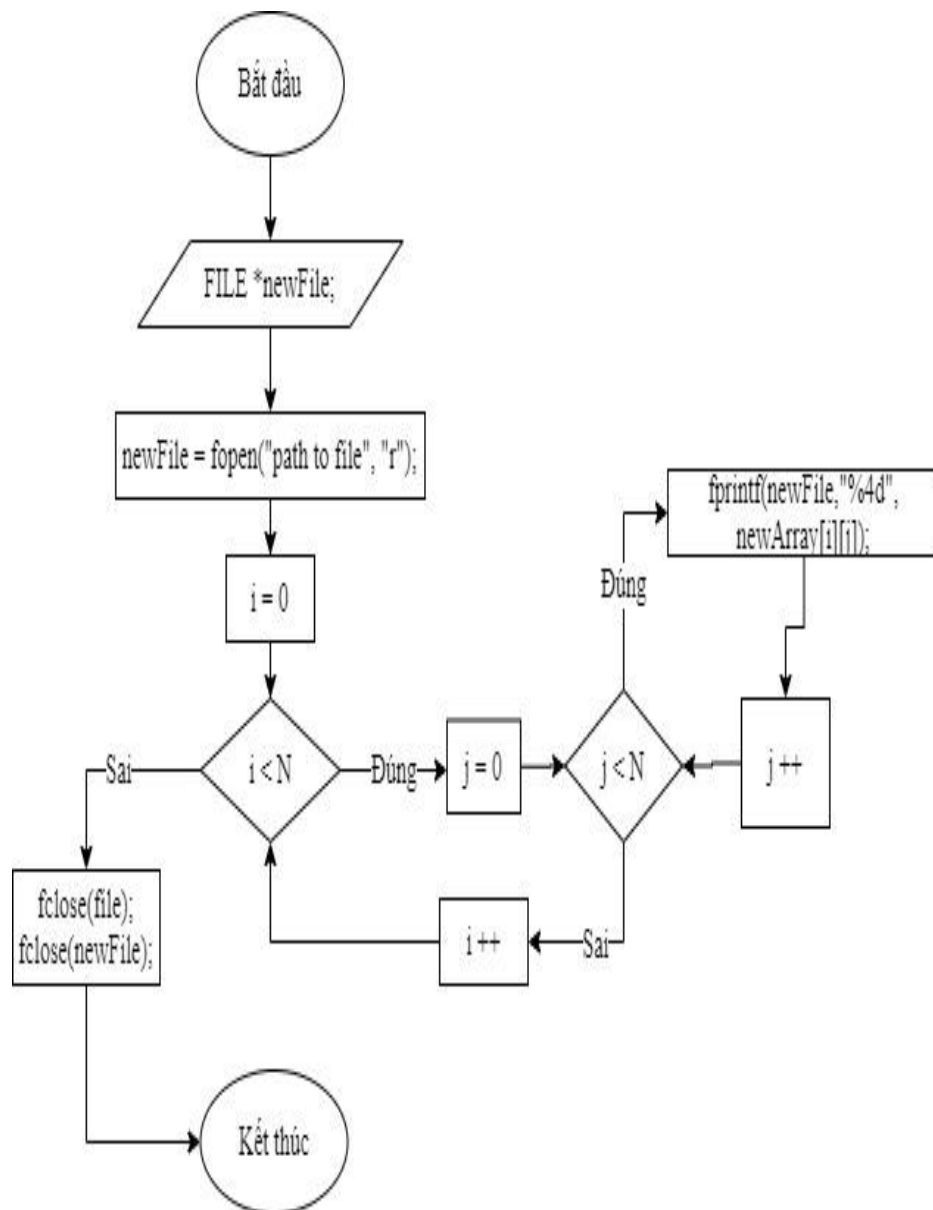
2.2.2 Lưu đồ thuật toán

Thuật toán để đọc file text vào mảng:



Hình 2.8 Thuật toán Nhập file vào mảng $a[N][N]$

Thuật toán để lưu mảng vào file đầu ra

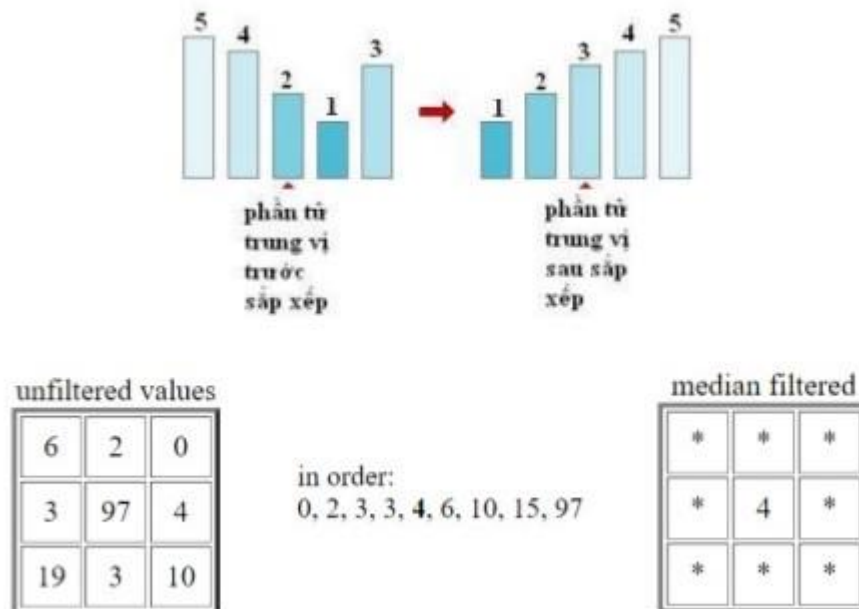


Hình 2.8 Thuật toán lưu mảng thành ma trận txt

2.3 Thuật toán làm mịn ảnh bằng lọc trung vị (Median filter)

2.3.1 Thuật toán

Ta sử dụng một ma trận lọc (ma trận 3×3) quét qua lần lượt từng điểm ảnh của ảnh đầu vào input. 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”. Sau đó sắp xếp các điểm ảnh trong từng cửa sổ này 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. Sẽ dễ hình dung ra hơn bằng ảnh dưới đây. [4]

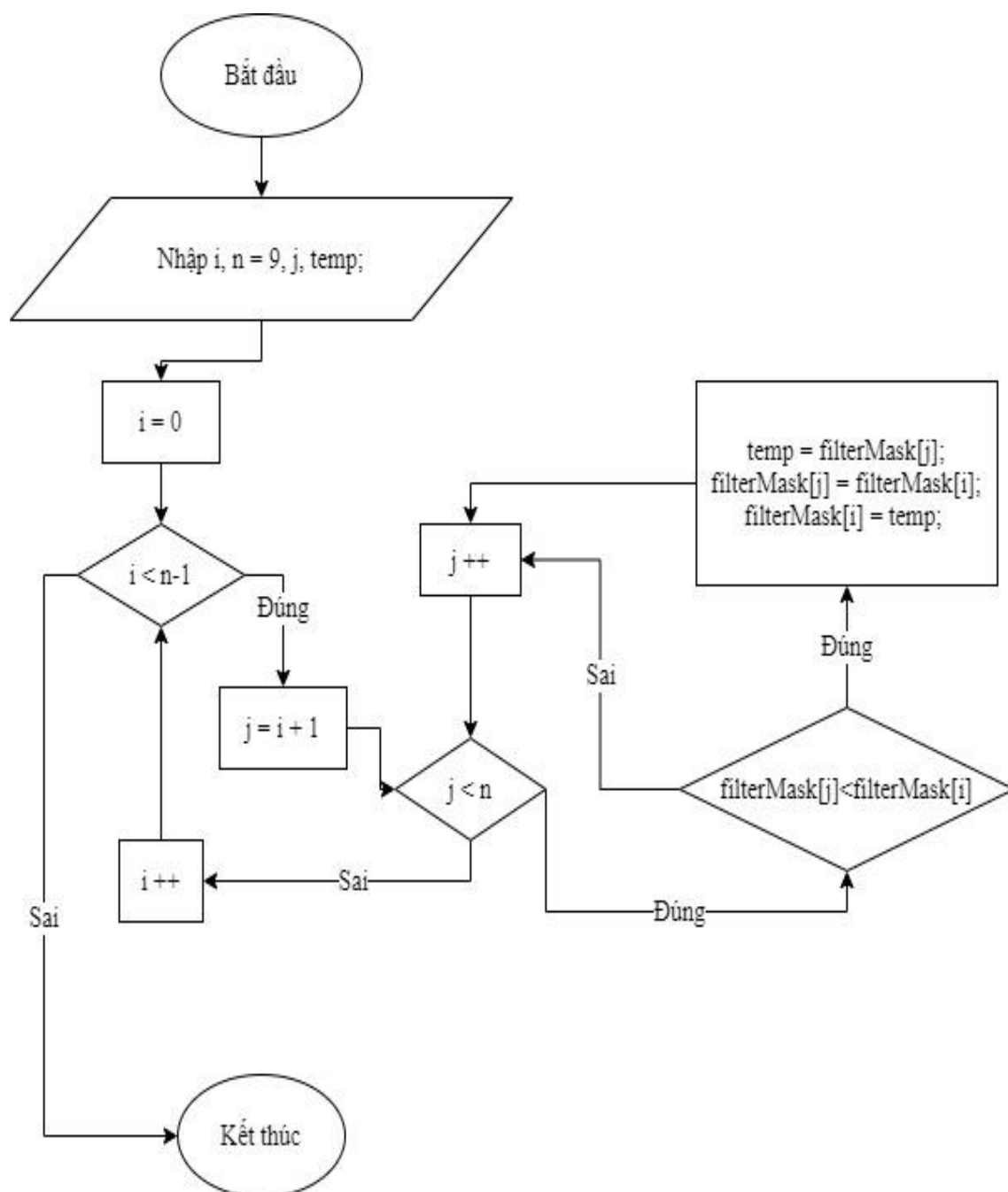


Hình 2.10 Mô tả thuật toán Median filter

2.3.2 Lưu đồ thuật toán

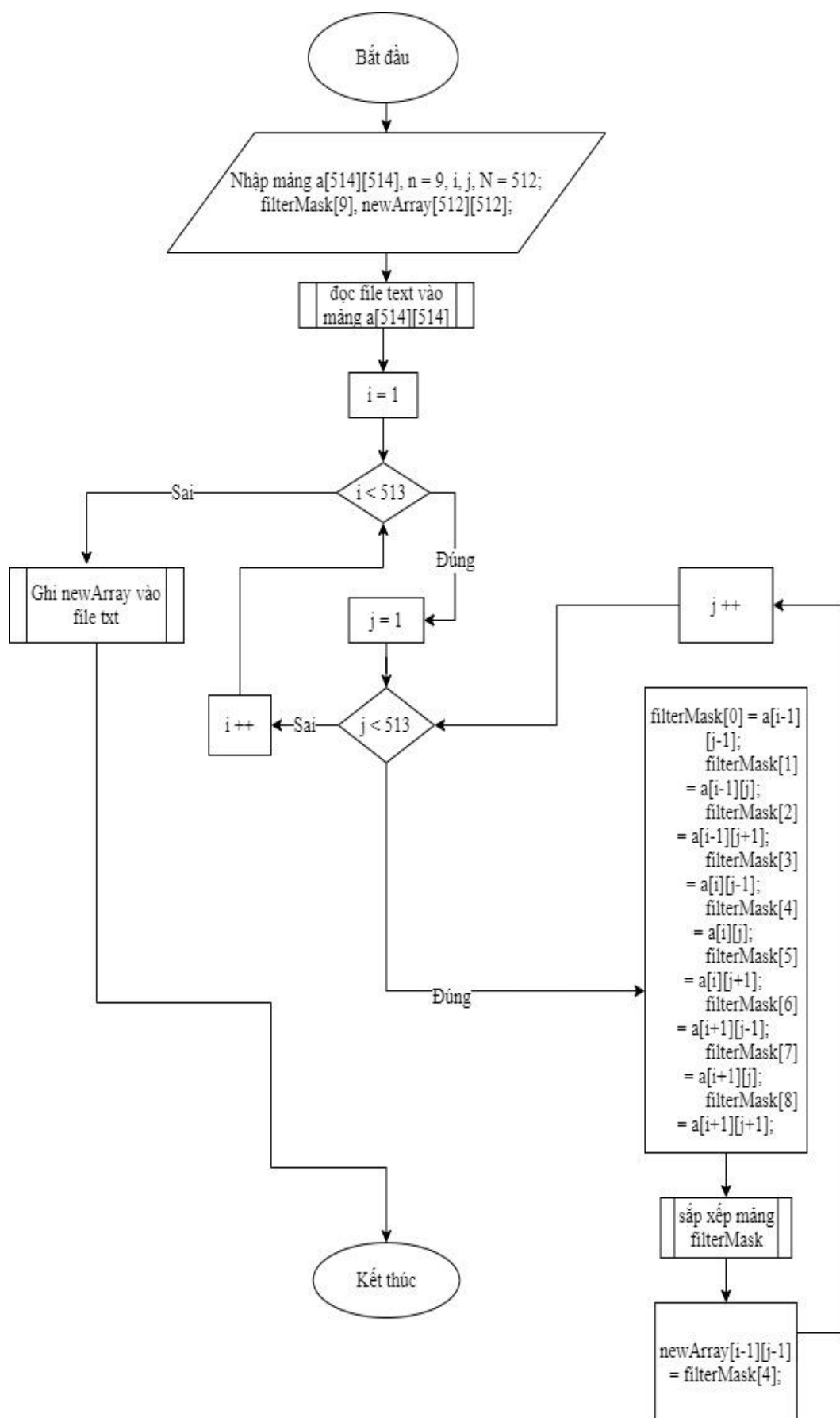
Thuật toán đọc và xuất file text tương tự với phần của histogram.

Thuật toán sắp xếp mảng: (Lưu đồ thuật toán được thực hiện bằng phần mềm vẽ lưu đồ online: <https://bietmaytinh.com/vesodo/>)



Hình 2.11 Thuật toán sắp xếp mảng

Thuật toán chính sử dụng bộ lọc Median Filter:



Hình 2.12 Thuật toán chính Median Filter

CHƯƠNG 3. THIẾT KẾ CẤU TRÚC DỮ LIỆU LƯU TRỮ ẢNH VÀ CÁC THÔNG SỐ LIÊN QUAN ĐẾN HAI THUẬT TOÁN TRÊN

3.1 Về chức năng cân bằng xám:

Trước khi thực hiện ta cần phải chuyển file text ma trận ảnh vào code, ta thực hiện lưu ma trận này vào mảng 2 chiều **a[N][N]** với N là height và width của ma trận, trong trường hợp của bài toán thì **N = 512**.

Trước hết phải lập ra histogram: ta tạo ra mảng **hist [256]**, trong đó các phần tử từ **hist [0]-hist [255]** tương ứng là các mức xám và các giá trị của **hist [0] - hist [255]** là số pixel dùng cho mức xám đó.

Để thuận tiện cho việc duyệt mảng để thực hiện thuật toán ta khai báo i, j, k.

Để tính toán cho hàm **cdf (v)** ta lập mảng **cdfHist [256]** với chức năng tương tự như **hist [256]**. Với mặc định giá trị của tất cả các phần tử của mảng ban đầu là 0 trừ **cdfHist [0] = hist [0]** để thuận tiện cho việc tính toán hàm **cdf (v)**.

Sau khi cân bằng ta lưu dữ liệu từ mảng **newArray**, ta dùng hàm ghi file text để lưu trữ ma trận ảnh đầu ra.

3.2 Về phép lọc trung vị

Tương tự như histogram equalization. Ta lưu trữ ma trận ảnh vào trong mảng **a [514][514]** . Tạo mảng như vậy để thiết lập các phần tử ở viền ma trận sẽ là 0 → thuận tiện hơn cho việc lọc trung vị.

Phục vụ cho việc lọc ảnh ta cần ma trận lọc 3 nhân 3 là mảng **filterMask [9]**.

Với các biến khai báo **i, j, n =9** để thuận tiện cho việc duyệt mảng để triển khai thuật toán. Và biến **temp** là phần tử trung gian phục vụ thuật toán sắp xếp để tìm ra phần tử trung vị.

Tạo mảng **newArray [512][512]** là ma trận ảnh đầu ra sau khi đã được lọc trung vị. Và dùng để ghi ra file text .

CHƯƠNG 4. MÃ NGUỒN TRIỂN KHAI HAI THUẬT TOÁN

4.1 Thuật toán lọc trung vị

```
#include <stdio.h>
#include <stdlib.h>
#define N 512

void sortArray(int filterMask[])
{
    int i,n = 9,j;
    int temp;
    for ( i = 0; i< n-1; i++)
    {
        for( j = i+1; j<n; j++)
        {
            if(filterMask[j]<filterMask[i])
            {
                temp = filterMask[j];
                filterMask[j] = filterMask[i];
                filterMask[i] = temp;
            }
        }
    }
}

int main()
{
    int a[514][514]; // array to represent image matrix
    FILE *f;
    int i,j;
    f = fopen("path_to_input_file","r");
    if( f == NULL)
    {
        printf("Error!");
        exit(1);
    }

    // Open file and read in into a 2D array.
    for( i = 0; i <514; i++)
    {
        a[i][0] = 0;
        a[i][513] = 0;
    }
    for( j = 0; j<514; j++)
    {
        a[0][j] = 0;
        a[513][j] = 0;
    }
}
```

```

    }

    for( i =1; i<513; i++)
    {
        for ( j=1; j<513; j++)
        {
            fscanf(f,"%d", &a[i][j]);
        }
    }

    // use Median Filter
    int filterMask[9];
    int newArray[512][512];
    // Median filter use a 3*3 filter matrix
    for (i =1; i<513; i++)
    {
        for(j =1; j<513; j++)
        {
            filterMask[0] = a[i-1][j-1];
            filterMask[1] = a[i-1][j];
            filterMask[2] = a[i-1][j+1];
            filterMask[3] = a[i][j-1];
            filterMask[4] = a[i][j];
            filterMask[5] = a[i][j+1];
            filterMask[6] = a[i+1][j-1];
            filterMask[7] = a[i+1][j];
            filterMask[8] = a[i+1][j+1];
            //sort maskArray:
            sortArray(filterMask);
            newArray[i-1][j-1] = filterMask[4]; // array for output matrix
        }
    }

    FILE *newFile;
    newFile = fopen("path_to_output_file","w");
    // write newArray into output file
    for( i =0; i<N; i++)
    {
        for ( j=0; j<N; j++)
        {
            fprintf(newFile,"%-4d", newArray[i][j]);
        }
        fprintf(newFile,"\n");
    }
    return 0;
}

```


4.2 Thuật toán cân bằng xám:

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#define N 512

int main()
{
    FILE *file;
    file = fopen("path_to_input_file", "r");//openfile
    if( file == NULL)
    {
        printf("Error!");
        exit(1);
    }

    int a[N][N], newArray[N][N], i, j, k;//array to represent image matrix
    int hist[256] = {0};
    int cdfHist[256] = {0};
    for ( i = 0; i<N; i++)
    {
        for ( j = 0; j<N; j++)
        {
            fscanf(file, "%d", &a[i][j]);
        }
        // Calculate for histogram (calculate frequency)
    }
    for ( k = 0; k< 256; k++)
    {
        for ( i = 0; i<N; i++)
        {
            for ( j = 0; j<N; j++)
            {
                if( a[i][j] == k)
                {
                    hist[k] += 1;
                }
            }
        }
    }

    //      The cumulative distribution function (cdf)

    cdfHist[0] = hist[0];
    for ( k = 1; k < 256; k++)
    {
        cdfHist[k] += (hist[k]+ cdfHist[k -1]);
        //      printf("\n cdfHist[%d] = %d",k, cdfHist[k]);
    }
```

```

// Calculate cdfmin
int cdfmin;
for( k = 0; k < 256; k++)
{
    if( hist[k] != 0)
    {
        cdfmin = cdfHist[k];
        break;
    }
}

// Final calculate for equalization histogram
for( k = 0; k < 256; k++)
{
    for( i = 0; i < N; i++)
    {
        for( j = 0; j < N; j++)
        {
            if( a[i][j] == k)
            {
                newArray[i][j] = ((cdfHist[k] -
cdfmin)*254)/(N*N- cdfmin);
            }
        }
    }
}

//write newArray into text file
FILE *newFile;
newFile = fopen("path_to_output_file","w");
for( i = 0; i < N; i++)
{
    for ( j = 0; j < N; j++)
    {
        fprintf(newFile,"%-4d", newArray[i][j]);
    }
    fprintf(newFile,"\n");
}

return 0;
}

```

CHƯƠNG 5. PHƯƠNG ÁN CHẠY THỬ THÍ NGHIỆM

5.1 Mô tả ảnh đầu vào

Ảnh đầu vào là ảnh xám kích thước tối đa 512×512 lưu dưới dạng file text. Trong đó dòng 1, 2, ..., 512 lưu giá trị độ xám (từ 0 đến 255) các điểm ảnh ở các cột 1, 2, ..., 512 của dòng 1, 2, ..., 512 tương ứng. Có độ tương phản kém hoặc bị nhiễu muối tiêu, nhiễu Gauss.

5.2 Cách thức chạy chương trình

Dùng phần mềm devC hoặc compiler code online repl.it/languages/c để thực hiện thuật toán và mã nguồn. Với file ảnh text đầu vào ta được đầu ra là ảnh được lưu dưới dạng file text. Tùy theo ảnh đầu nếu nhiễu ta sẽ thực hiện median filter còn nếu tương phản kém ta sẽ thực hiện cân bằng histogram, điều này có thể dễ dàng nhận biết qua mắt thường hoặc dùng matlab để có thể theo dõi được histogram của ảnh.

5.3 Phương án đánh giá ảnh đầu ra

Bằng ứng dụng MATLAB ta có thể chuyển đổi file ma trận text đầu ra thành ảnh lưu dưới dạng .png hoặc tùy chọn để theo dõi kết quả.

Qua đó ta có thể đánh giá kết quả dễ dàng bằng mắt thường (bằng cách so sánh 2 ảnh ban đầu và ảnh sau khi dùng bộ lọc). Ngoài ra ta có thể so sánh ma trận đầu ra và histogram sau khi cân bằng với ảnh và histogram được tính toán chuẩn bằng MATLAB hoặc cũng có thể dễ dàng nhìn ra bằng mắt thường.

5.4 Kết quả thử nghiệm trên các tệp ảnh đầu vào

Sau đây là một số ví dụ thử nghiệm trên các tệp ảnh đầu vào và được chuyển ra file ảnh theo code trong MATLAB

Code trình bày lọc trung vị:

```
grayImage = uint8(importdata(path_to_text_file));  
imwrite (grayImage, 'path_to_image_file');% convert matrix to image  
% read image
```

```

I = imread('path_to_image_file_after_change');

img = imread(' path_to_image_file_before_change");

% show 2 image to compare

subplot(1,2,1);

imshow(img);

subplot(1,2,2);

imshow(I);

```

Code trình bày cân bằng xám

```

grayImage = uint8(importdata(path_to_text_file));

imwrite (grayImage, 'path_to_image_file');% convert matrix to image.

% read image

I = imread('path_to_image_file_after_change');

img = imread(' path_to_image_file_before_change");

% show 2 image to compare

subplot(1,4,1);

imshow(img);

subplot(1,4,2);

imshow(I);

subplot(1,4,3);

imhist(img); % show histogram

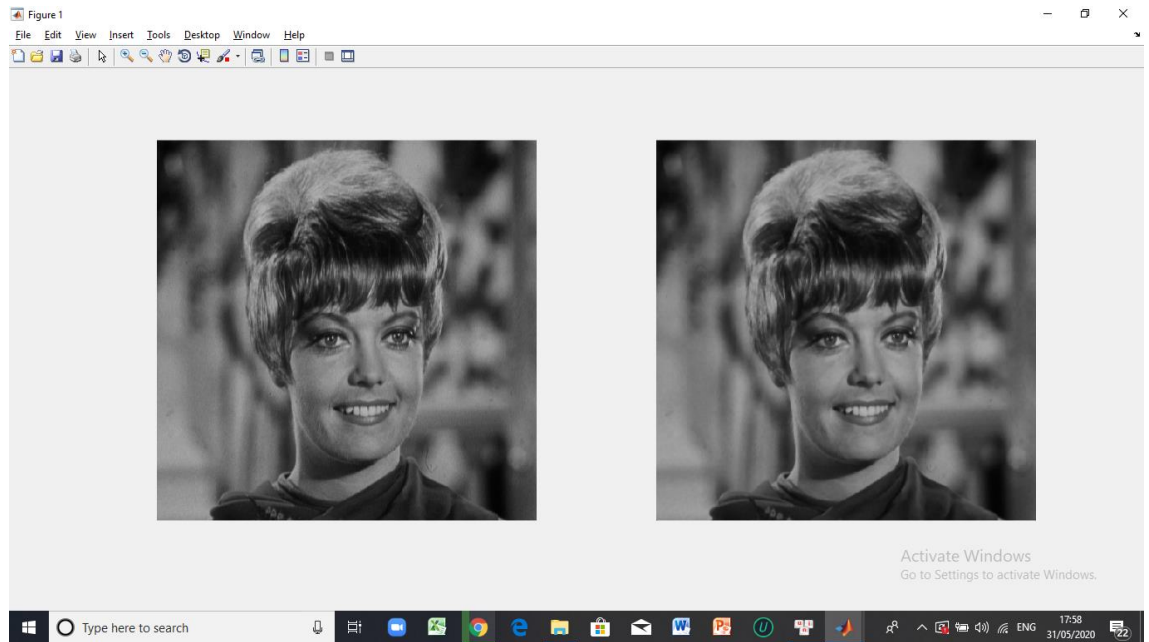
subplot(1,4,4);

imhist(I);

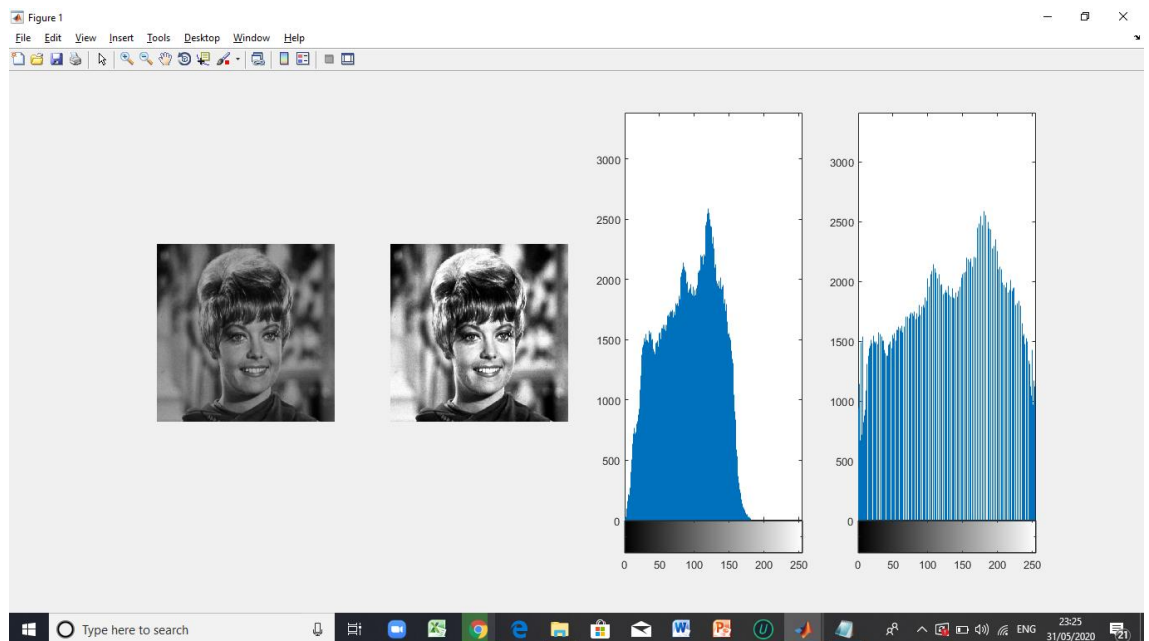
```

So sánh ảnh ban đầu (bên trái) với ảnh đã qua bộ lọc trung vị (bên phải). Ta thấy ảnh qua bộ lọc có phần mịn hơn và nếu zoom lên có thể dễ dàng thấy đã loại bỏ hiệu quả những đốm nhiễu.

Đối với ảnh qua bộ lọc cân bằng xám, ta có thể dễ dàng thấy bằng mắt thường chất lượng, độ tương phản của ảnh đã cải thiện rõ rệt. Ngoài ra ta có thể so sánh biểu đồ xám, ta thấy biểu đồ xám sau khi được cân bằng sẽ có các giá trị mức xám được dẫn đều.

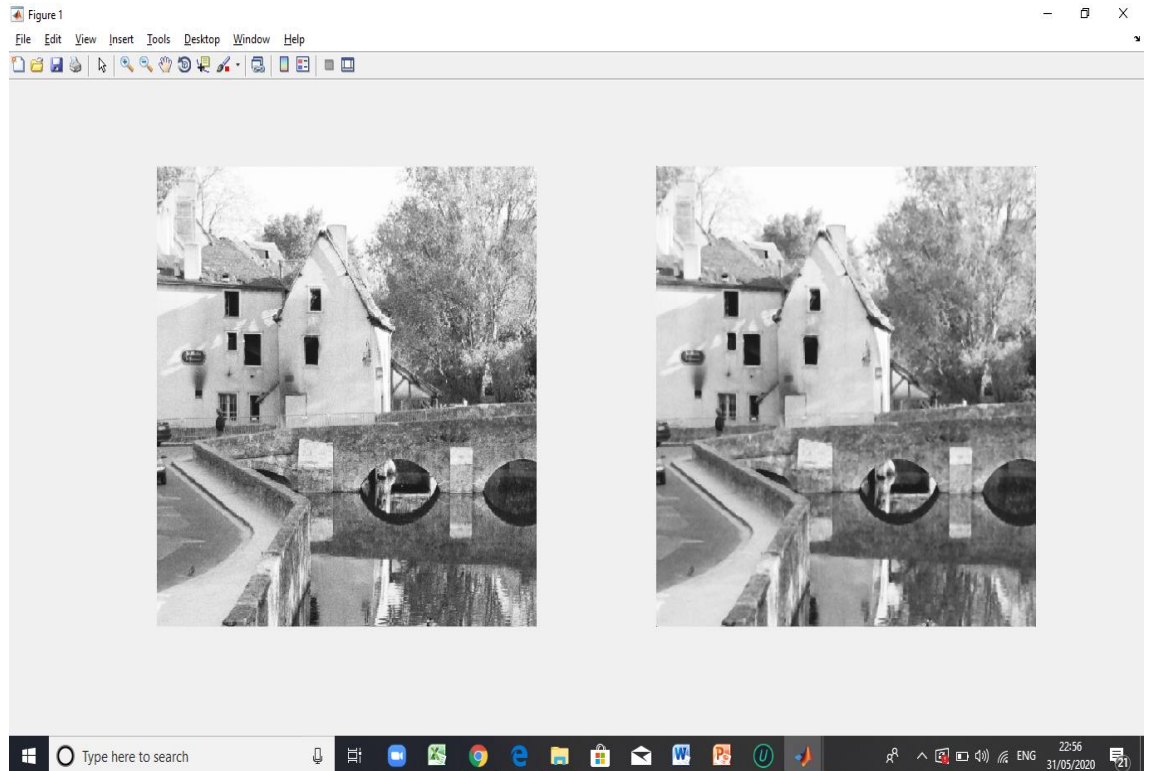


Hình 5.1 So sánh ảnh ban đầu và ảnh qua bộ lọc trung vị

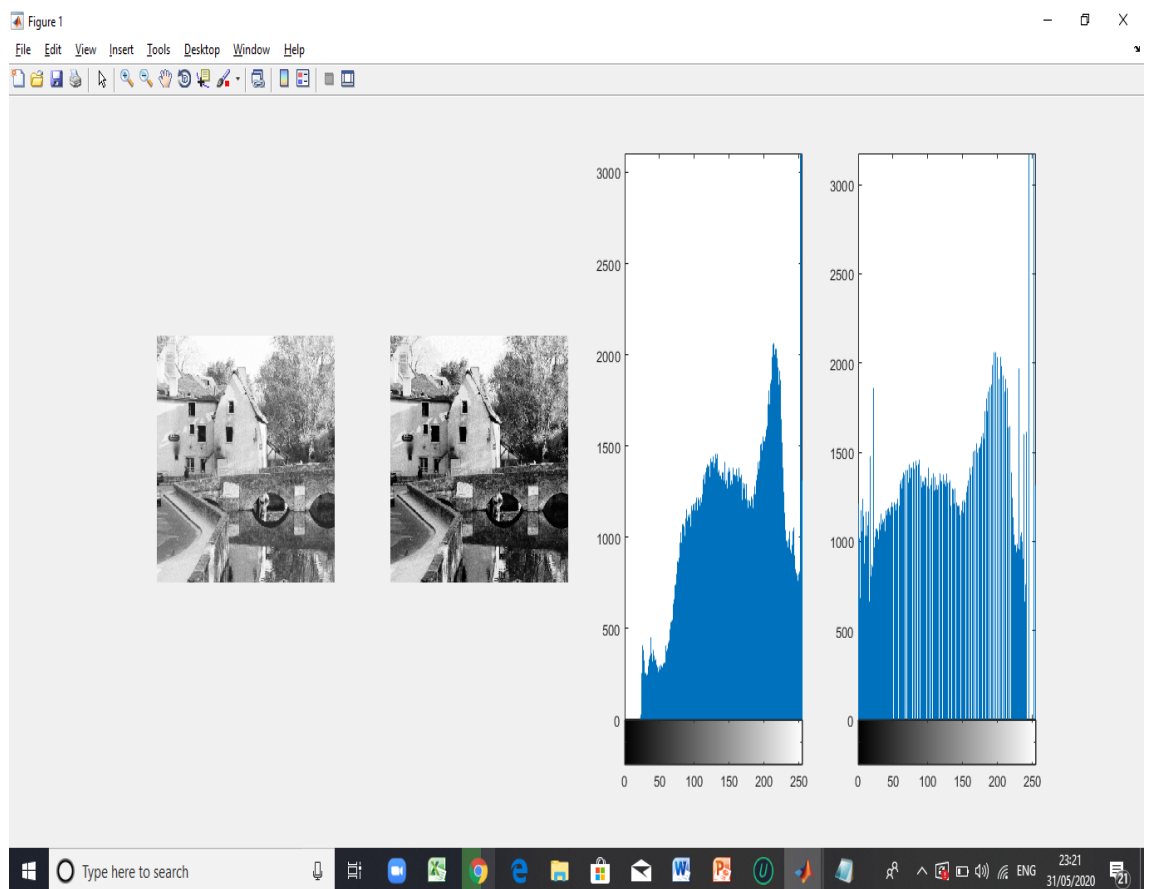


Hình 5.2 So sánh ảnh ban đầu và ảnh qua bộ lọc cân bằng xám

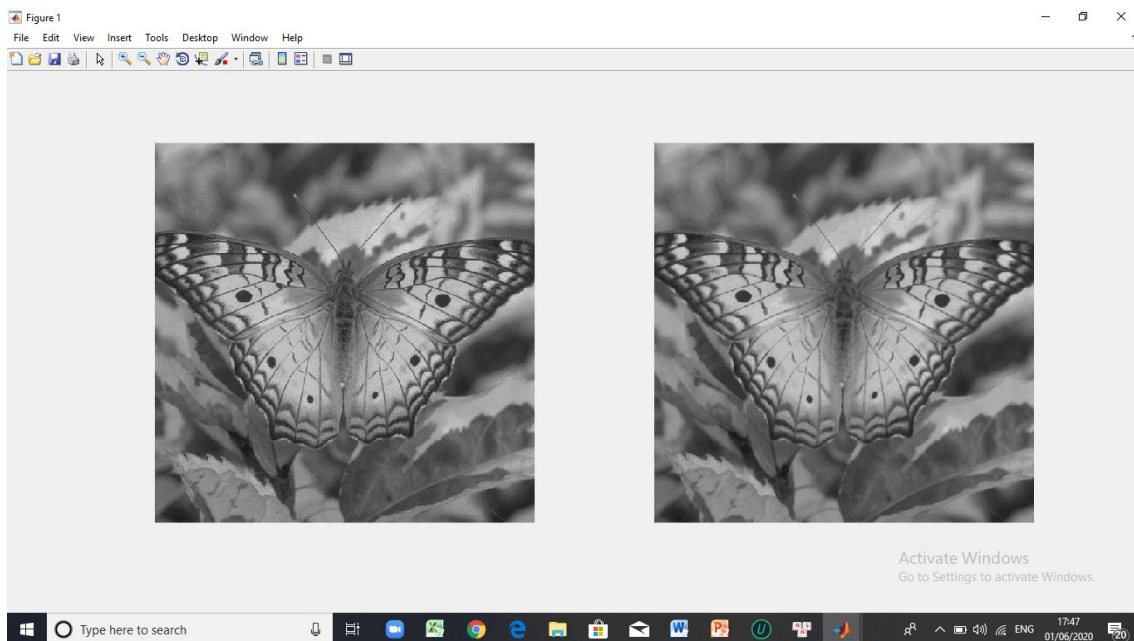
Đối với các ví dụ tiếp theo chúng ta xem kết quả mà 2 bộ lọc làm được:



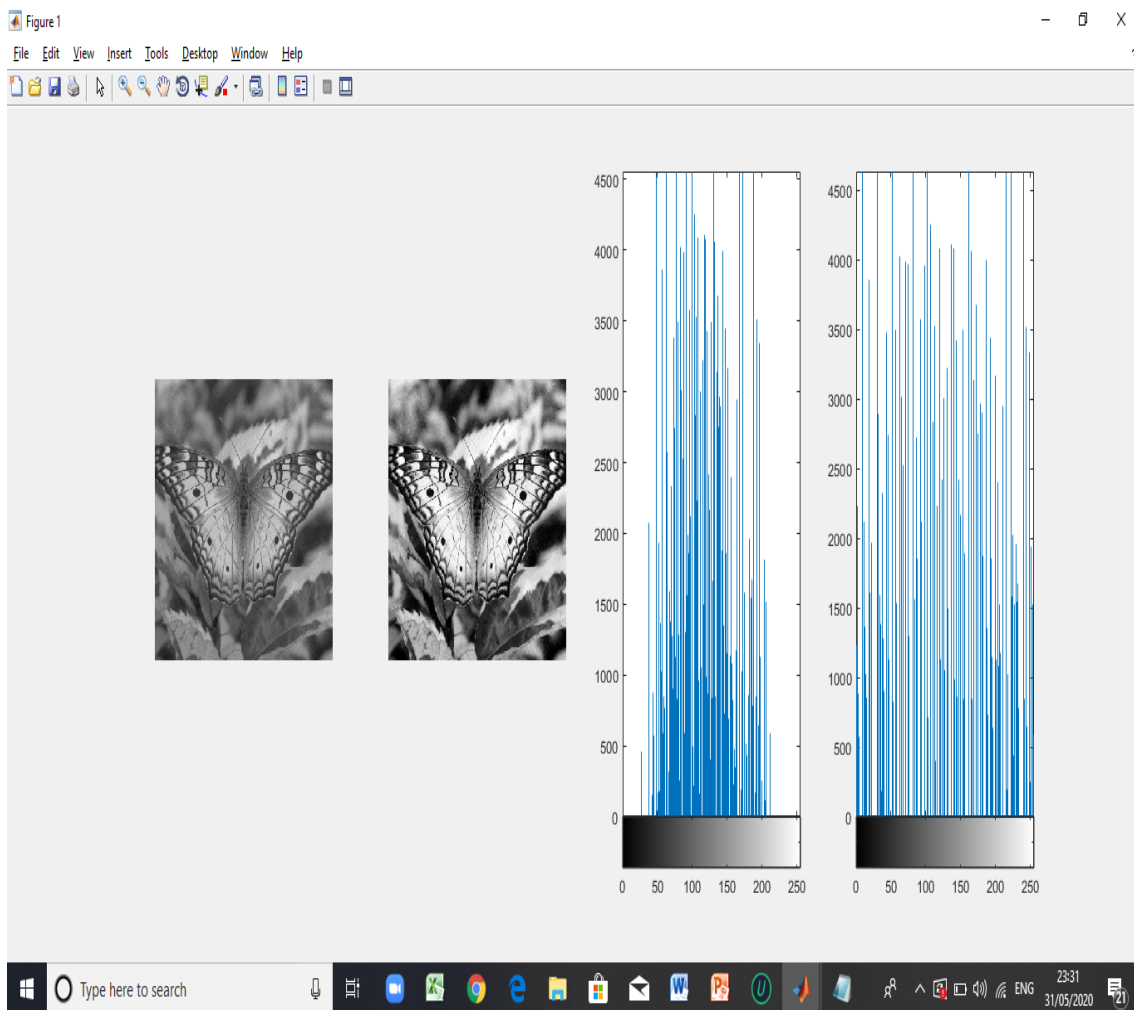
Hình 5.3



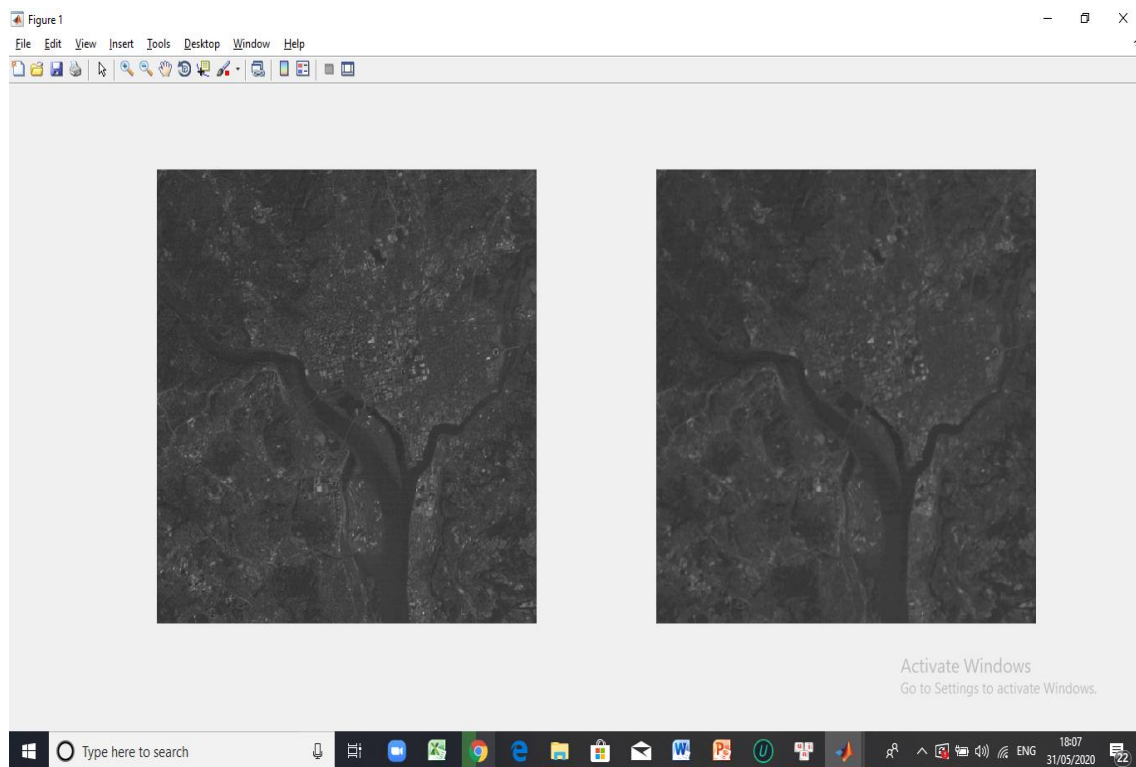
Hình 5.4



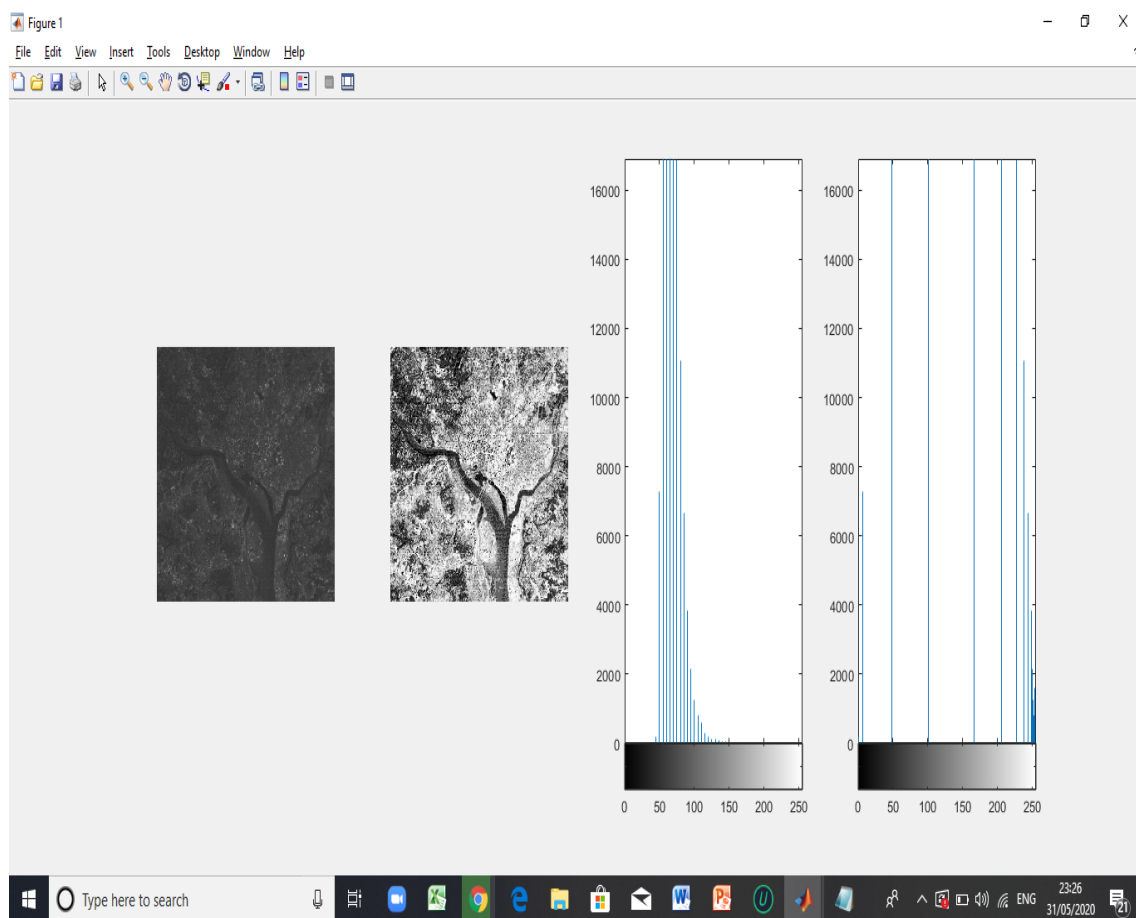
Hình 5.5



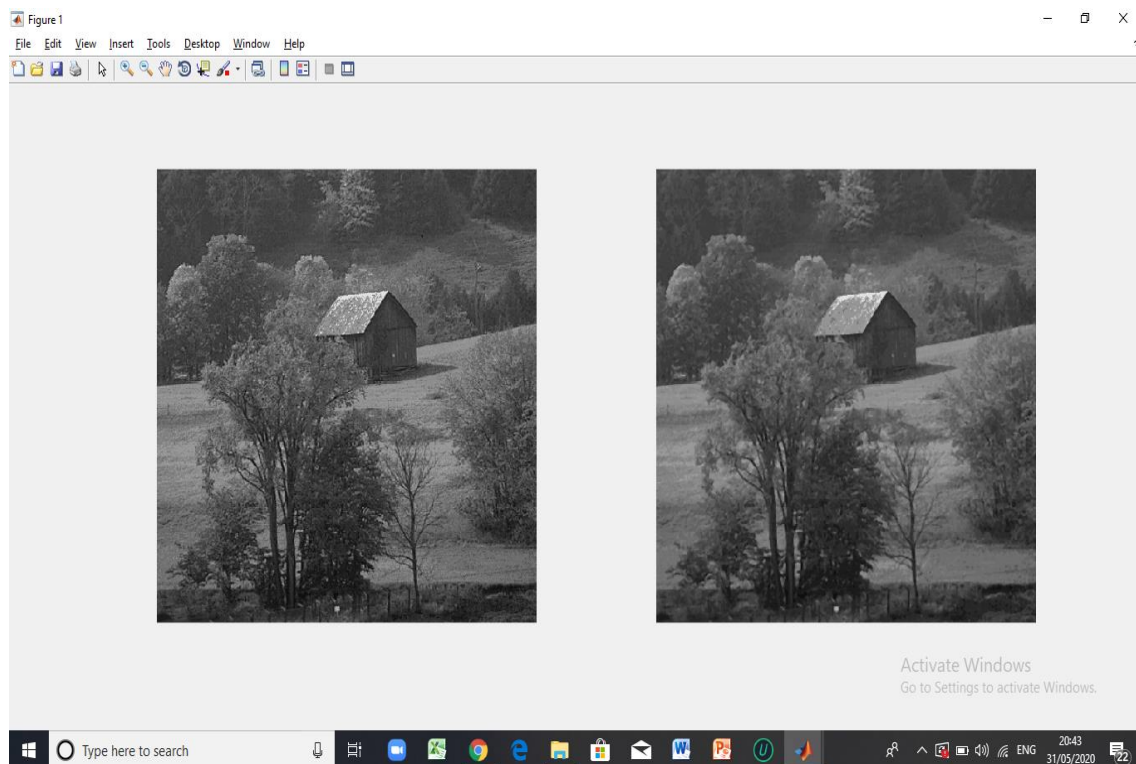
Hình 5.6



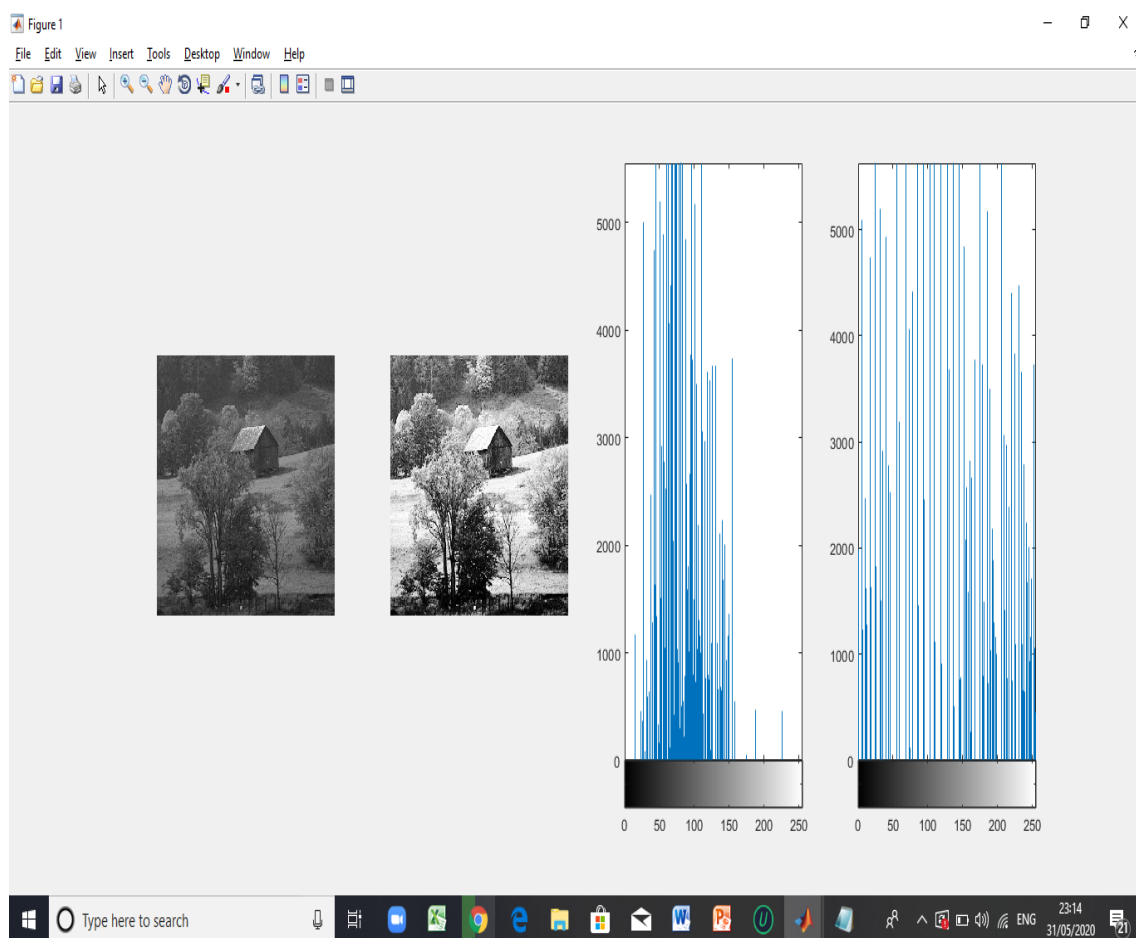
Hình 5.7



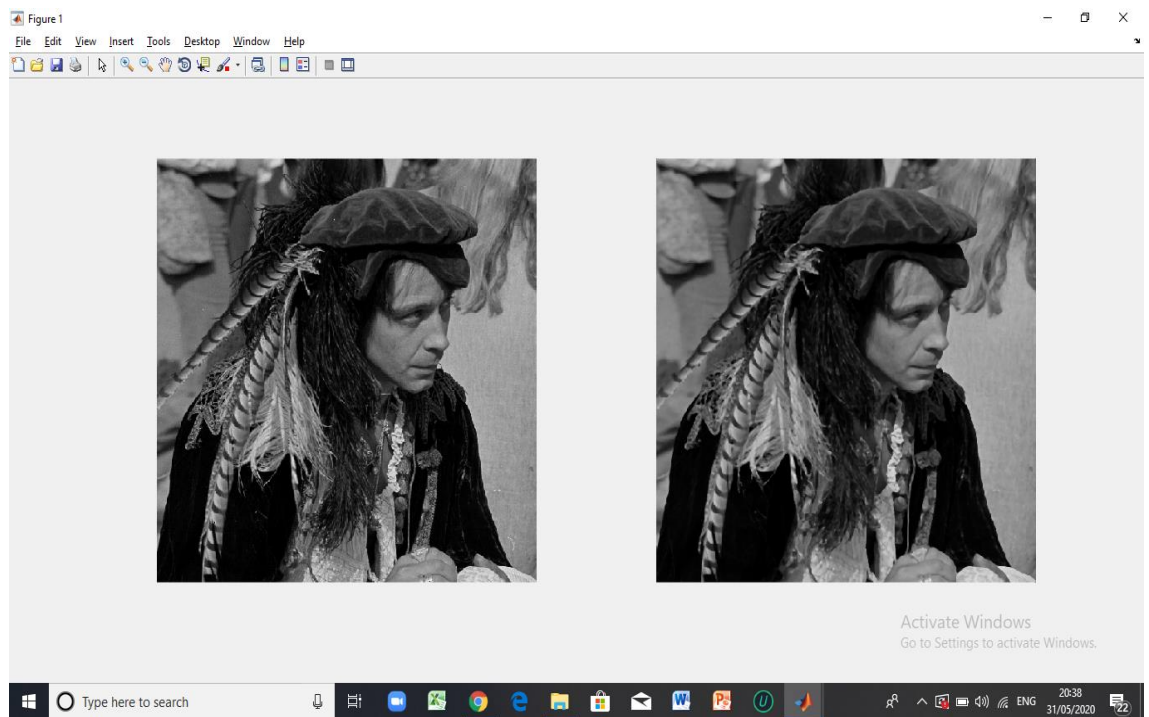
Hình 5.8



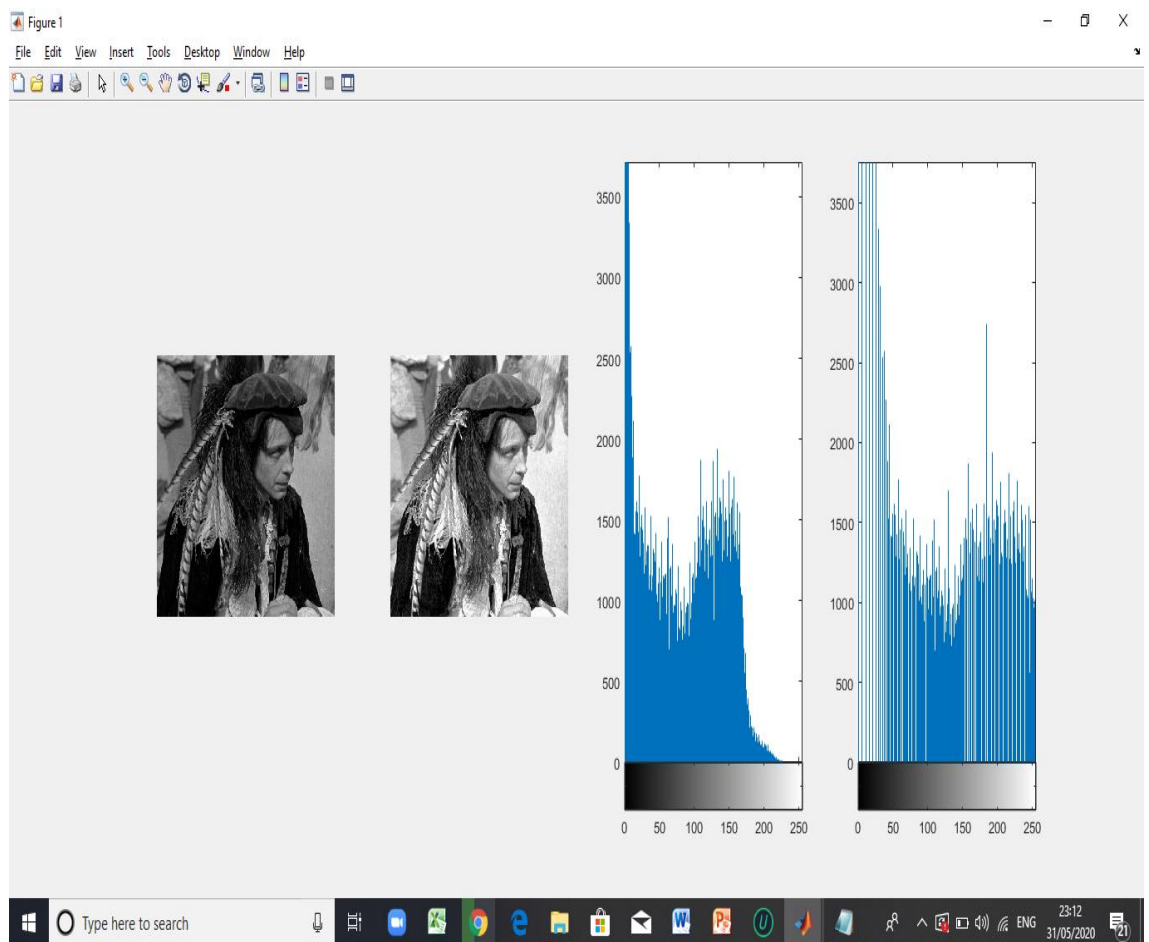
Hình 5.9



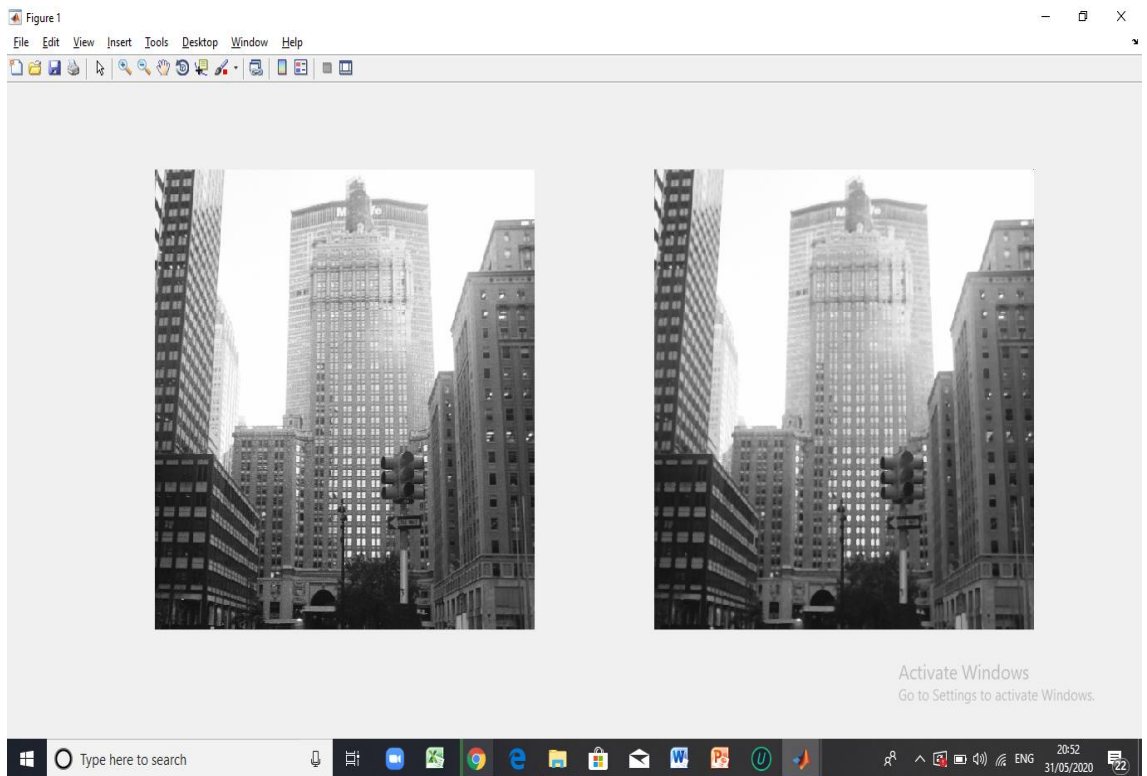
Hình 5.10



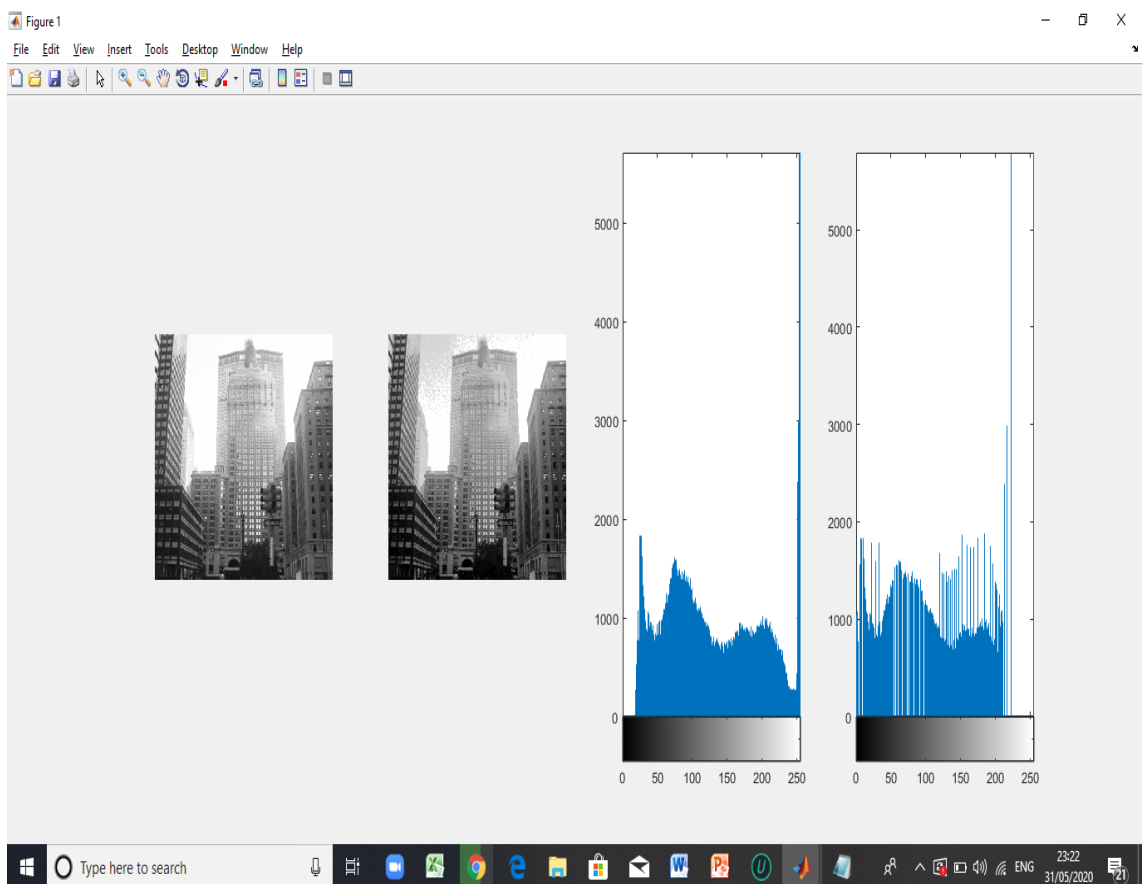
Hình 5.11



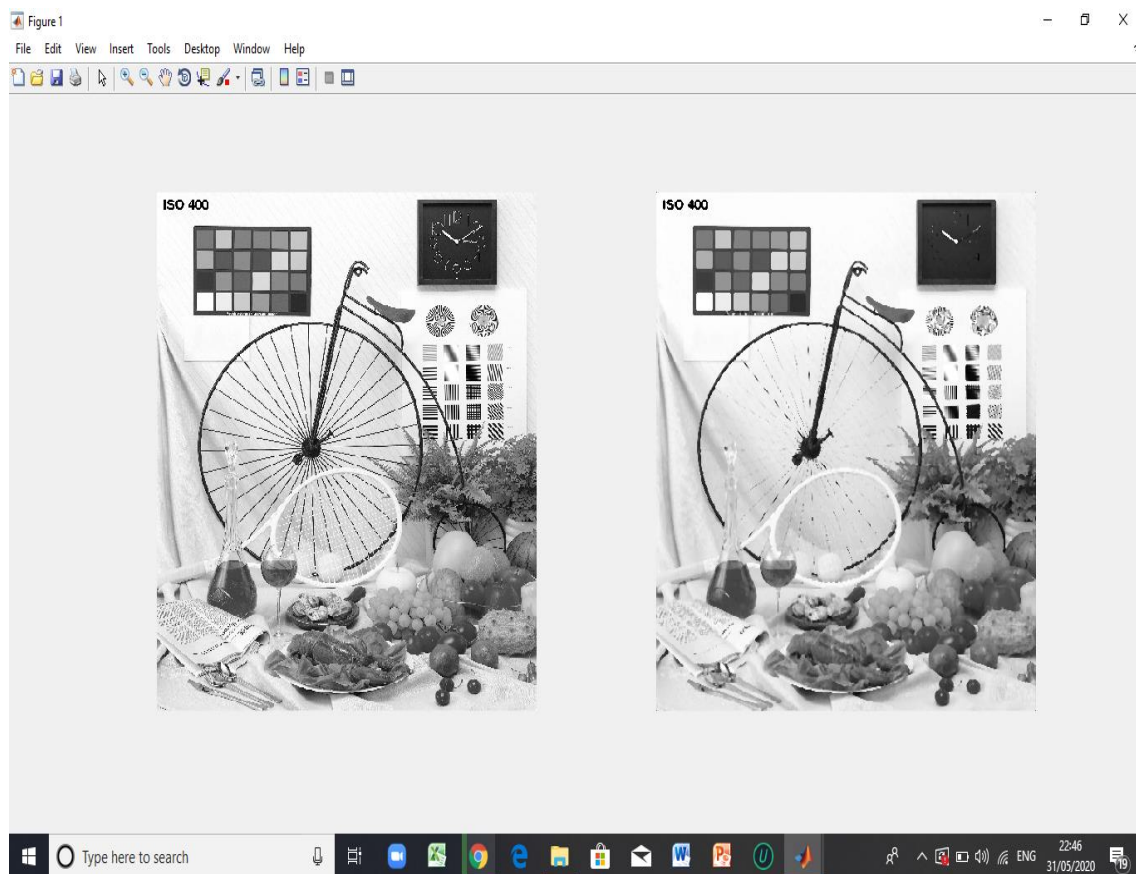
Hình 5.12



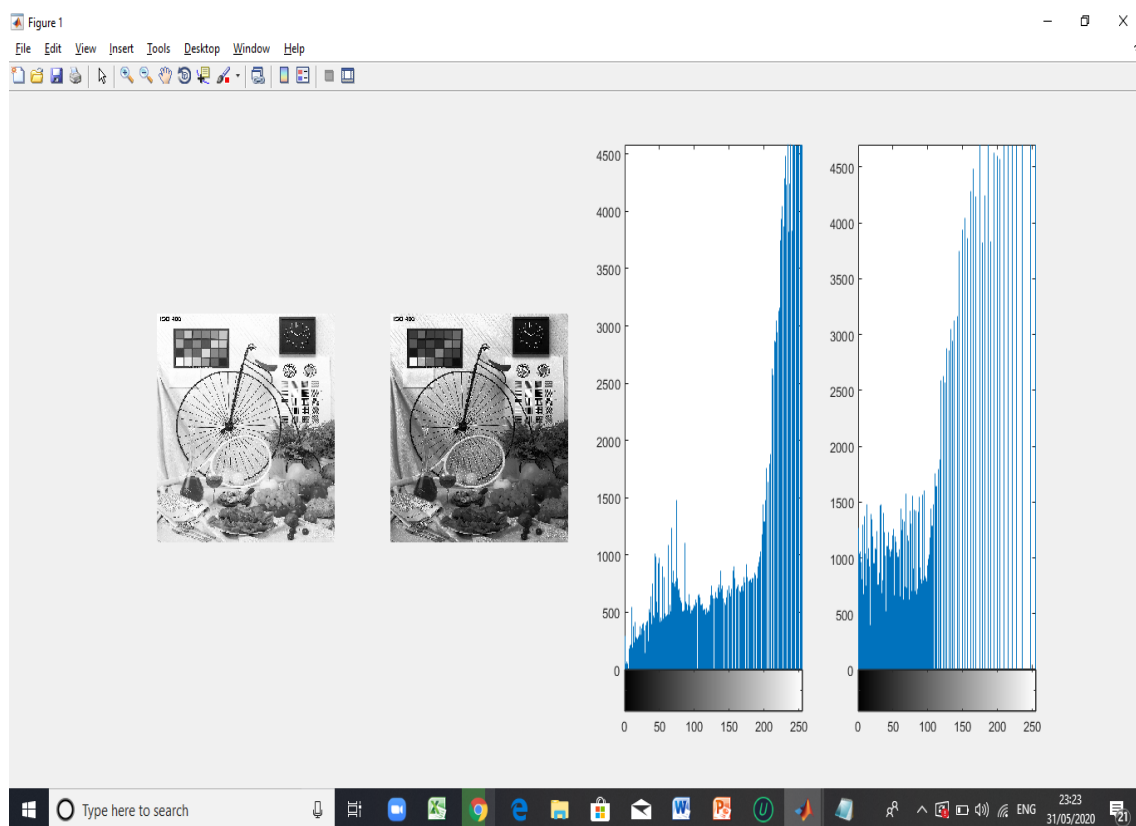
Hình 5.13



Hình 5.14



Hình 5.15



Hình 5.16

CHƯƠNG 6. KẾT LUẬN

6.1 Kết luận

Báo cáo đã đáp ứng được yêu cầu của đề tài, đã chuyển đổi được ma trận ảnh qua bộ lọc và cho ra kết quả như yêu cầu của đề bài. Các kết quả thu được cho thấy các ảnh cần được thực hiện phương pháp nâng cao chất lượng ảnh phù hợp với trạng thái nhiễu, mức độ tương phản qua đó thực hiện lọc trung vị hoặc cân bằng xám hoặc kết hợp cả hai để cho ra được ảnh với chất lượng nét cũng như độ tương phản tốt nhất.

Đề tài có tính ứng dụng thực tế cao, được ứng dụng rộng rãi trong nhiều lĩnh vực đời sống và đóng vai trò quan trọng như trong lĩnh vực y tế (làm nét ảnh siêu âm hay chụp x –quang, phát hiện thân nhiệt bất thường), trong lĩnh vực an ninh quân sự (theo dõi trong môi trường ánh sáng kém, thu ảnh từ vệ tinh hay radar). Ngoài ra còn ứng dụng tốt trong lĩnh vực nhiếp ảnh, phục chế ảnh.

Qua đề tài này, em đã học được nhiều kỹ năng bổ ích cho bản thân như giải quyết vấn đề, tìm hiểu thông tin, phân bổ thời gian thực hiện công việc và cải thiện hơn kỹ năng trình bày, kỹ năng viết báo cáo khoa học kỹ thuật. Những kỹ năng này sẽ giúp em tốt hơn trong tương lai và trong con đường học tập.

TÀI LIỆU THAM KHẢO

[1] Median filter, Wikipedia, Web page:

https://en.wikipedia.org/wiki/Median_filter

[2] Luận văn đề tài làm tron ảnh, Phạm Việt Thắng, Web page:

<https://www.slideshare.net/trongthuy1/luan-van-tim-hieu-bai-toan-lam-tron-anh-hay?ref=https://tailieumau.vn/luan-van-tot-nghiep-tim-hieu-bai-toan-lam-tron-anh-hay/>

[3] Histogram equalization, Wikipedia, Web page:

https://en.wikipedia.org/wiki/Histogram_equalization

[4] Bài giảng Xử lý ảnh, cô Hồ Thị Hương Thơm, Web page:

https://www.youtube.com/watch?v=qpxgR4cpJQg&list=PL_jAnFPw7i2GikC2K8hcgjFsqXv2xhQAM

[5] Bài giảng Xử lý ảnh, giảng viên Trần Đức Quang- Viện CNTT và Truyền Thông- ĐHBK Hà Nội, Web page:

<https://users.soict.hust.edu.vn/ductq/XLA%20Lecture.pdf>

[6] CVG-UGR Image Dataset, Webpage:

<http://decsai.ugr.es/cvg/dbimagenes/g512.php>

[7] File ảnh text đầu vào, Web page:

https://drive.google.com/drive/folders/1XTDVKmk3WGaXVXbTym15XA7UEg_qnGRT

PHỤ LỤC

1. File ảnh sau đầu ra đã được chuyển đổi từ file text → file ảnh .png:
<https://drive.google.com/drive/folders/1cMFCC2i7rTLQ353iyRDrRhBFbSpJVdtN>
2. File code:
<https://drive.google.com/drive/folders/1tTuQdbT4DoJyZAeC50q49Nmp7yUIuG0n>
3. Link compiler online:
Media Filter: <https://repl.it/join/kaocfxrf-snhong3>
Histogram Equalization: <https://repl.it/join/ptxfuvco-snhong3>

