TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN &

TRUYỀN THÔNG VIỆT HÀN

**Khoa Khoa Học Máy Tính**



ĐỒ ÁN CƠ SỞ 5

**XÂY DỰNG CHƯƠNG TRÌNH   
XỬ LÝ ẢNH**

Sinh viên thực hiện : **Lê Thành An**

**Lê Văn Hiểu**

Lớp : **19IT2**

Giảng viên hướng dẫn : **TS. Lê Thị Thu Nga**

***Đà Nẵng, tháng 05 năm 2022***

**NHẬN XÉT**

(của giảng viên hướng dẫn)

∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙∙

*Đà Nẵng, Ngày… Tháng… Năm 2022*

**Giảng viên hướng dẫn**

**TS.Lê Thị Thu Nga**

**LỜI CẢM ƠN**

Để đồ án này đạt kết quả tốt đẹp, em đã nhận được sự hỗ trợ, giúp đỡ của thầy cô. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả quý thầy cô đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu và phát triển đề tài.

Trước hết em xin gửi tới các thầy cô trường Đại Học Công Nghệ Thông Tin Và Truyền Thông Việt Hàn-Đại Học Đà Nẵng lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đồ án.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới cô – TS.Lê Thị Thu Nga đã quan tâm, giúp đỡ tận tình em hoàn thành tốt đồ án này trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án 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 kiến thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc104914565)

[CHƯƠNG 1 CƠ SỞ LÝ THUYẾT 2](#_Toc104914566)

[1.1 Khái quát về xử lý ảnh 2](#_Toc104914567)

[1.1.1 Một số khái niệm cơ bản 2](#_Toc104914568)

[1.1.2 Một số ứng dụng trong xử lý ảnh 2](#_Toc104914569)

[1.1.3 Định nghĩa về histogram 3](#_Toc104914570)

[1.2 Xử lý nâng cao chất lượng ảnh 5](#_Toc104914571)

[1.2.1 Khái niệm và phân loại các kiểu cải thiện ảnh 5](#_Toc104914572)

[1.2.2 Cải thiện ảnh sử dụng các toán tử điểm 6](#_Toc104914573)

[1.2.3 Cải thiện ảnh sử dụng các toán tử không gian 9](#_Toc104914574)

[1.3 Khôi phục ảnh 16](#_Toc104914575)

[1.3.1 Nhiễu và mô hình nhiễu 16](#_Toc104914576)

[1.3.2 Các loại nhiễu 16](#_Toc104914577)

[1.3.3 Các kỹ thuật lọc nhiễu 17](#_Toc104914578)

[1.4 Không gian màu 19](#_Toc104914579)

[1.4.1 Không gian màu HSV 19](#_Toc104914580)

[1.4.2 Không gian màu RBG 20](#_Toc104914581)

[1.4.3 Không gian màu CMYK 21](#_Toc104914582)

[CHƯƠNG 2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG 22](#_Toc104914583)

[2.1 Mô tả yêu cầu 22](#_Toc104914584)

[2.2 Ngôn ngữ lập trình và các thư viện hỗ trợ 22](#_Toc104914585)

[2.3 Phân tích 23](#_Toc104914586)

[2.3.1 Biểu đồ Usecase 23](#_Toc104914587)

[2.3.2 Đặc tả usecase 23](#_Toc104914588)

[2.3.3 Biểu đồ hoạt động 30](#_Toc104914589)

[CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH XỬ LÝ ẢNH 31](#_Toc104914590)

[3.1 Xây dựng giao diện 31](#_Toc104914591)

[3.2 Xây dựng chức năng 31](#_Toc104914592)

[3.2.1 Chức năng đảo ngược mức sáng 31](#_Toc104914593)

[3.2.2 Chức năng phân ngưỡng 32](#_Toc104914594)

[3.2.3 Chức năng điều chỉnh độ tương phản 32](#_Toc104914595)

[3.2.4 Chức năng lọc ảnh 32](#_Toc104914596)

[3.2.5 Chức năng cân bằng histogram 33](#_Toc104914597)

[KẾT LUẬN 34](#_Toc104914598)

[TÀI LIỆU THAM KHẢO 35](#_Toc104914599)

**DANH MỤC HÌNH**

[Hình 1.1 Histogram tối 3](#_Toc104914600)

[Hình 1.2 Histogram của ảnh quá tối 4](#_Toc104914601)

[Hình 1.3 Histogram của ảnh quá sáng 4](#_Toc104914602)

[Hình 1.4 Histogram của ảnh quá tương phản 4](#_Toc104914603)

[Hình 1.5 Phân loại các phương pháp cải thiện ảnh 6](#_Toc104914604)

[Hình 1.6 Biến đổi hàm mũ với c=1 γ=0.6 7](#_Toc104914605)

[Hình 1.7 Biến đổi âm bản 7](#_Toc104914606)

[Hình 1.8 Biến đổi tăng độ tương phản (a), phân ngưỡng (b) 8](#_Toc104914607)

[Hình 1.9 Biến đổi cắt ngưỡng 9](#_Toc104914608)

[Hình 1.10 Biến đổi Logarit 9](#_Toc104914609)

[Hình 1.11 Bộ lọc trung bình 10](#_Toc104914610)

[Hình 1.12 Bộ lọc trung bình có trọng số 11](#_Toc104914611)

[Hình 1.13 Lọc trung vị 7x7 12](#_Toc104914612)

[Hình 1.14 Ảnh lọc Gaussian kích thước 21 x 21 khi K = 1; **σ** =3.5 13](#_Toc104914613)

[Hình 1.15 Bộ loc Sobel 14](#_Toc104914614)

[Hình 1.16 Bộ lọc Robert cross gradient 14](#_Toc104914615)

[Hình 1.17 Bộ lọc Laplacian 15](#_Toc104914616)

[Hình 1.18 Ví dụ ảnh sử dụng bộ lọc Laplacian 15](#_Toc104914617)

[Hình 1.19 Các biến thể khác nhau của Laplacian 15](#_Toc104914618)

[Hình 1.20 Mô hình nhiễu 16](#_Toc104914619)

[Hình 1.21 Lọc đảo khôi phục ảnh nguyên 17](#_Toc104914620)

[Hình 1.22 Thành phần Hue 20](#_Toc104914621)

[Hình 2.1 Biểu đồ usecase 23](#_Toc104914622)

[Hình 2.2 Biểu đồ hoạt động 30](#_Toc104914623)

[Hình 3.1 Giao diện phần mềm 31](#_Toc104914624)

[Hình 3.2 Điều chỉnh độ tương phản 32](#_Toc104914625)

[Hình 3.3 Lọc Gaussian 33](#_Toc104914626)

[Hình 3.4 Cân bằng Histogram 33](#_Toc104914627)

**DANH MỤC BẢNG**

[Table 2‑1 Đặc tả usecase mở ảnh 23](#_Toc104914628)

[Table 2‑2 Đặc tả usecase zoom ảnh 24](#_Toc104914629)

[Table 2‑3 Đặc tả usecase thêm nhiễu 25](#_Toc104914630)

[Table 2‑4 Đặc tả usecase lọc sobel 26](#_Toc104914631)

[Table 2‑5 Đặc tả usecase phân vùng ảnh 26](#_Toc104914632)

[Table 2‑6 Đặc tả usecase cần bằng Histogram 27](#_Toc104914633)

[Table 2‑7 Đặc tả usecase lọc gaussian 28](#_Toc104914634)

LỜI MỞ ĐẦU

Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh được đưa vào giảng dạy ở bậc đại học ở nước ta khoảng chục năm nay. Nó là môn học liên quan đến nhiều lĩnh vực và cần nhiều kiến thức cơ sở khác. Đầu tiên phải kể đến Xử lý tín hiệu số là một môn học hết sức cơ bản cho xử lý tín hiệu chung, các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn…Thứ hai, các công cụ toán như Đại số tuyến tính, sác xuất thống kê. Một số kiến thứ cần thiết như Trí tuệ nhân tạo, Mạng nơron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh.

Mục tiêu chính của xử lý ảnh là việc tinh chỉnh xử lý hình ảnh phục vụ cho nhu cầu giải trí, nghiên cứu và trao đổi thông tin ngày càng cao của con người.

Ngoài ra công nghệ AI cũng đang từng bước phát triển và được ứng dụng ngày càng nhiều vào đời sống hàng ngày. Lĩnh vực xử lý ảnh cũng đang được ứng dụng nhiều công nghệ AI vào quá trình xử lý.

Đề tài của chúng em bao gồm 3 chương:

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích và thiết kế hệ thống

Chương 3: Xây dựng chương trình xử lý ảnh

# CƠ SỞ LÝ THUYẾT

## Khái quát về xử lý ảnh

### Một số khái niệm cơ bản

Ảnh có thể xem là tập hợp các điểm ảnh và mỗi điểm ảnh được xem như là đặc trưng cường độ sáng hay một dấu hiệu nào đó tại một vị trí nào đó của đối tượng trong không gian và nó có thể xem như một hàm n biến P(c, c1, c2, …). Do đó, ảnh trong xử lý ảnh có thể xem như ảnh n chiều.

**\* Ảnh và điểm ảnh:**

Điểm ảnh được xem như là dấu hiệu hay cường độ sáng tại một vị trí nào đó của đối tượng trong không gian và ảnh được xem như là một tập hợp các điểm ảnh.

**\* Mức xám, màu**

Là số các giá trị có thể có của các điểm ảnh của ảnh

### Một số ứng dụng trong xử lý ảnh

Như đã nói ở trên, các kỹ thuật xử lý ảnh trước đây chủ yếu được sử dụng để nâng cao chất lượng hình ảnh, chính xác hơn là tạo cảm giác về sự gia tăng chất lượng ảnh quang học trong mắt người quan sát. Thời gian gần đây, phạm vi ứng dụng xử lý ảnh mở rộng không ngừng, có thể nói hiện không có lĩnh vực khoa học nào không sử dụng các thành tựu của công nghệ xử lý ảnh số.

Trong y học các thuật toán xử lý ảnh cho phép biến đổi hình ảnh được tạo ra từ nguồn bức xạ X -ray hay nguồn bức xạ siêu âm thành hình ảnh quang học trên bề mặt film x-quang hoặc trực tiếp trên bề mặt màn hình hiển thị. Hình ảnh các cơ quan chức năng của con người sau đó có thể được xử lý tiếp để nâng cao độ tương phản, lọc, tách các thành phần cần thiết (chụp cắt lớp) hoặc tạo ra hình ảnh trong không gian ba chiều (siêu âm 3 chiều).

Trong lĩnh vực địa chất, hình ảnh nhận được từ vệ tinh có thể được phân tích để xác định cấu trúc bề mặt trái đất. Kỹ thuật làm nổi đường biên (image enhancement) và khôi phục hình ảnh (image restoration) cho phép nâng cao chất lượng ảnh vệ tinh và tạo ra các bản đồ địa hình 3-D với độ chính xác cao.

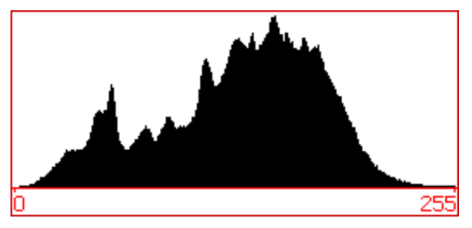
Trong ngành khí tượng học, ảnh nhận được từ hệ thống vệ tinh theo dõi thời tiết cũng được xử lý, nâng cao chất lượng và ghép hình để tạo ra ảnh bề mặt trái đất trên một vùng rộng lớn, qua đó có thể thực hiện việc dự báo thời tiết một cách chính xác hơn.

### Định nghĩa về histogram

Lược đồ mức xám (histogram) của một ảnh, từ nay về sau ta qui ước gọi là lược đồ xám, là một hàm cung cấp tần suất xuất hiện của mỗi mức xám (grey level). Lược đồ xám được biểu diễn trong một hệ toạ độ vuông góc x,y. Trong hệ toạ độ này, trục hoành biểu diễn số mức xám từ 0 đến N, N là số mức xám (256 mức trong trường hợp chúng ta xét). Trục tung biểu diễn số điểm ảnh cho một mức xám (số điểm ảnh có cùng mức xám). Cũng có thể biểu diễn khác một chút: trục tung là tỷ lệ số điểm ảnh có cùng mức xám trên tổng số điểm ảnh.

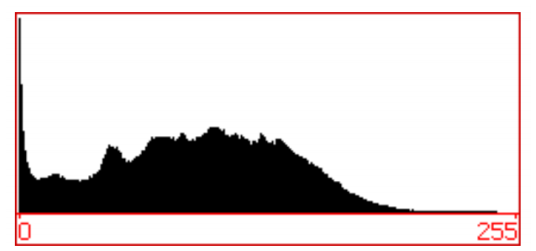
Histogram cung cấp cho những thông cơ bản, như độ sáng và độ tương phản (contrast) của ảnh. Độ tương phản đặc trưng cho sự thay đổi độ sáng của đối tượng so với nền. Có thể nói, độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền. Ta có một vài nhận xét về histogram:

+ NX1. Histogram tốt có hình ngọn núi với độ cao tăng dần từ trái, cao nhất ở giữa và thấp nhất ở bên phải. Điều đó chứng tỏ số lượng điểm ảnh nhiều nhất là ở độ sáng trung bình.



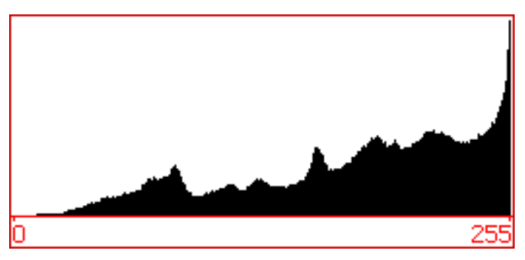
Hình 1.1 Histogram tối

+ NX2. Ảnh quá tối: histogram bị nghiêng về bên trái, có một cái cột gần như thẳng đứng sát trái.



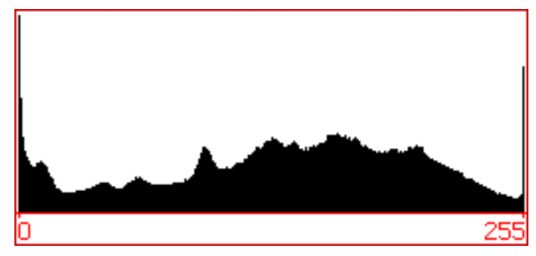
Hình 1.2 Histogram của ảnh quá tối

+ NX3. Ảnh quá sáng: histogram bị nghiêng về bên phải, có một cái cột gần như thẳng đứng sát phải.



Hình 1.3 Histogram của ảnh quá sáng

+ NX4. Ảnh quá tương phản: có hai cái cột nằm ở 2 đầu trái phải



Hình 1.4 Histogram của ảnh quá tương phản

+ NX5. Ảnh kém tương phản: dải màu bị dồn vào giữa, hai đầu không có gì.

Từ lược đồ xám ta có thể suy diễn ra các tính chất quan trọng của ảnh như giá trị xám trung bình hoặc độ tản mạn. Qua cách tác động lên điểm ảnh, sự phân bố của biểu đồ cột được thay đổi theo mục đích. Dựa vào lược đồ xám chúng ta có thể xác định được ngưỡng thích hợp cho quá trình phân đoạn hoặc tính được các đại lượng đặc trưng của một ảnh.

## Xử lý nâng cao chất lượng ảnh

### Khái niệm và phân loại các kiểu cải thiện ảnh

Cải thiện chất lượng ảnh:

* Tăng cường các đặc điểm của ảnh về mặt cảm nhận hoặc cục bộ như các đường nét, đường biên, hay độ tương phản, màu sắc, ..., lọc nhiễu.
* Tăng cường có hiệu quả cho các bước xử lý tiếp theo như hiển thị ảnh hoặc phân tích ảnh.
* Xử lý cải thiện ảnh chất lượng ảnh không làm tăng thông tin vốn có chứa trong dữ liệu.
* Làm tăng dải động của các thuộc tính của ảnh. Những thuộc tính này giúp cho phân biệt dễ dàng các chi tiết trên ảnh.

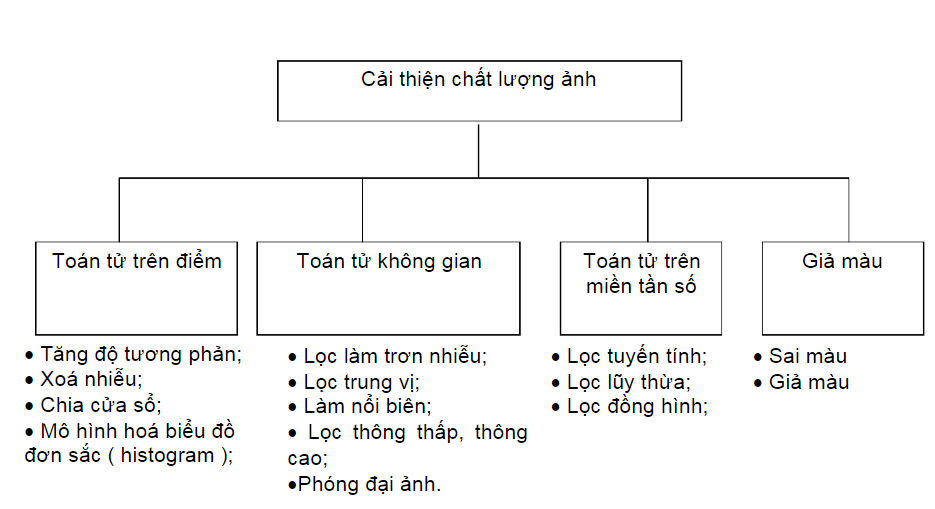
Các phương pháp cải thiện chất lượng ảnh bao gồm:

* Biến đổi phân bố mức xám , thay đổi độ tương phản;
* Giảm nhiễu, làm nổi biên và làm trơn biên của ảnh, lọc ảnh;
* Đưa thông tin vào ảnh;
* Phóng đại, thu nhỏ ảnh;
* Giả màu, …

Khó khăn: xác định các tiêu chuẩn định tính và định lượng về chất lượng ảnh;

Các kỹ thuật cải thiện chất lượng ảnh phụ thuộc vào kinh nghiệm xử lý;

Phân loại các phương pháp cải thiện chất lượng ảnh



Hình 1.5 Phân loại các phương pháp cải thiện ảnh

### Cải thiện ảnh sử dụng các toán tử điểm

Các phép toán không phụ thuộc không gian là các phép toán không phục thuộc vị trí của điểm ảnh. Ví dụ: Phép tăng giảm độ sáng, phép thống kê tần suất, biến đổi tần suất v.v..

Một trong những khái niệm quan trọng trong xử lý ảnh là biểu đồ tần suất (Histogram): Biểu đồ tần suất của mức xám g của ảnh I là số điểm ảnh có giá trị g của ảnh I. Ký hiệu là h(g)

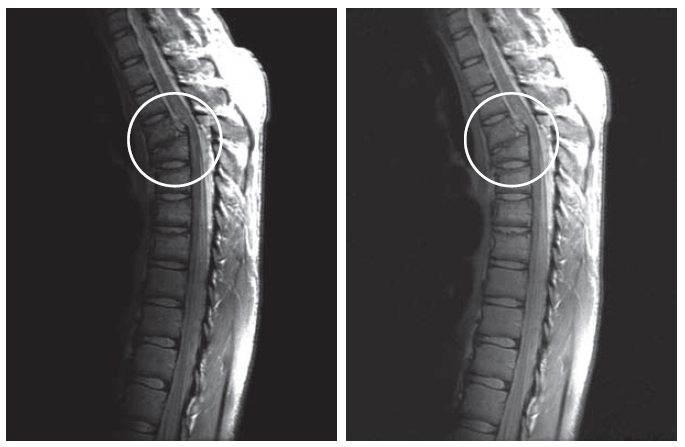
#### Tăng giảm độ sáng

Có 2 cách để tăng giảm độ sáng

Cách 1: sử dụng hàm số mũ: (gamma)

Là thao tác xử lý điểm ảnh trên ảnh có dạng

**s = c × rγ**



Hình 1.6 Biến đổi hàm mũ với c=1 γ=0.6

Với s là điểm ảnh đã xử lý, r là điểm ảnh đầu vào, c, γ hằng số dương

Cách 2: Chuyển hình ảnh về hệ màu HSV sau đố chỉnh giá trị h của pixel

Sẽ trình bày ở phần sau

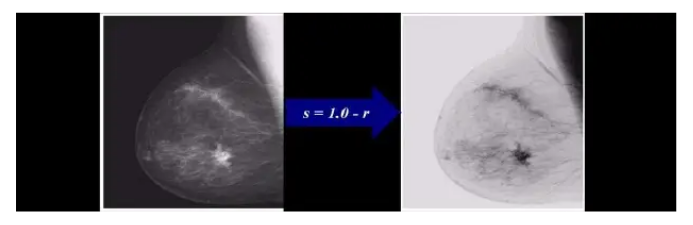
#### Biến đổi đảo ngược ảnh (âm bản)

**s =1.0 – r**

Hay

**s = Cường độ sáng lớn nhất − r**

Ảnh âm bản có ích trong việc tăng cường chi tiết màu trắng hoặc màu xám nhúng trong vùng tối của ảnh. Ví dụ biến đổi âm bản trên hình 10.



Hình 1.7 Biến đổi âm bản

#### Biến đổi tăng độ tương phản

Biến đổi tương phản tức là làm cho các vùng tối của ảnh sẽ trở nên tối hơn, vùng sáng sẽ trở nên sáng hơn hoặc ngược lại. Khi tang độ tương phản, nhìn vào biểu đồ histogram của ảnh ta sẽ thấy đồ thị cao dầu về phía 2 đầu của trục, khi giảm độ tương phản ta sẽ thấy histogram cao lên ở giữ như ngọn núi.

Để biến đổi độ tương phản ta sử dụng công thức:

g(x) = α f(x) + β với α >0

Trong đó:

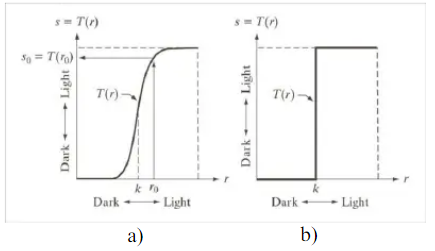
α và β để kiểm soát độ tương phản

f(x) là hình ảnh đầu vào

g(x) là hình ảnh đầu ra

Nếu α < 1 và β > 0 thì làm giảm tương phản của ảnh

Nếu α > 1 và β < 0 thì làm tăng tương phản của ảnh



Hình 1.8 Biến đổi tăng độ tương phản (a), phân ngưỡng (b)

#### Biến đổi cắt ngưỡng

Đưa ảnh đang có nhiều mức xám (cường độ sáng các màu red, geen, blue) về thành ảnh nhị phân nếu ảnh là ảnh xám, nhị phân 3 kênh red, green, blue. Trong đó, các pixel/giá trị của các kênh lớn hơn ngưỡng được đưa về bằng mức sáng cao nhất (thường là 1 hoặc 255). Các pixel còn lại bằng 0

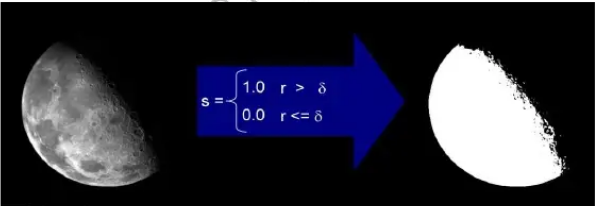
1, r > δ

S =

0, r <= δ

Trong đó δ là ngưỡng.

Biến đổi phân ngưỡng đặc biệt hữu ích để phân đoạn trong đó chúng ta muốn cô lập một đối tượng quan tâm từ nền của ảnh. Ví dụ biến đổi phân ngưỡng trên hình 11 với δ =0.25



Hình 1.9 Biến đổi cắt ngưỡng

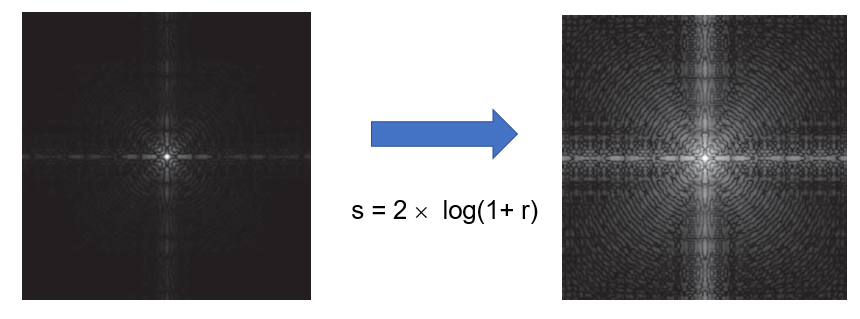
#### Biến đổi Logarit

Là thao tác xử lý điểm ảnh trên ảnh có dạng

**s = c × log(1+ r)**

Với s là điểm ảnh đã xử lý, r ≥ 0 là điểm ảnh đầu vào, c hằng số

Việc chuyển đổi logarit ánh xạ phạm vi hẹp của mức xám ngõ vào có giá trị thấp thành phạm vi rộng hơn của giá trị ngõ ra. Biến đổi logarit ngược thực hiện ngược lại.



Hình 1.10 Biến đổi Logarit

### Cải thiện ảnh sử dụng các toán tử không gian

Cơ sở lọc trong miền không gian

* Sử dụng mặc nạ lọc có kích thước: 3 × 3, 4 × 4, …
* Mặc nạ lọc di chuyển trên ảnh và thao tác lân cận với điểm ảnh
* Các thao tác lân cận đơn giản: Tính min, max, trung bình, trung vị, …

Việc lọc có thể được biểu diễn dạng hàm như sau:



Trong đó:

g(x,y) : là ảnh sau khi lọc

f(x,y) : là ảnh đầu vào

w(s,t): mặt nạ lọc

a = (m-1)/2 và b = (n-1)/2

m x n : là kích thước bộ lọc

Các kỹ thuật lọc không gian:

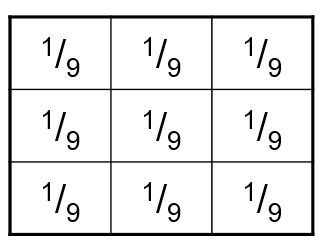
* Kỹ thuật lọc mịn ảnh (lọc thông thấp)
* Lọc trung bình
* Lọc trung bình có trọng số
* Lọc trung vị
* Bộ lọc Gausse
* Kỹ thuật lọc sắc nét ảnh (lọc thông cao)
* Bộ lọc đạo hàm bậc 1, bậc 2
* Bộ lọc Laplace, Sobel, Robert cross gradient
* Kết hợp các bộ lọc

#### Kỹ thuật lọc mịn ảnh

1. **Lọc trung bình**

Chỉ đơn giản tính trung bình các pixel lân cận của pixel trung tâm

Sử dụng loại bỏ nhiễu, làm nỗi các chi tiết lớn



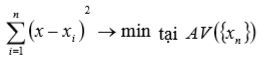
Hình 1.11 Bộ lọc trung bình

**Định nghĩa (Trung bình)**

Cho dãy x1, x2…, xn khi đó trung bình của dãy ký hiệu AV({xn}) được định nghĩa:

[](https://2.bp.blogspot.com/-_oWvWfH0v58/W8BvRbCmxUI/AAAAAAAABmQ/JUmrcY_xip8rRhhcqizaZ_0IH0Ny-pNCwCLcBGAs/s1600/image030.png)

**Mệnh đề**

[](https://3.bp.blogspot.com/-fwI958fg5kY/W8BvYBv8MBI/AAAAAAAABmU/x1DhatkbsZkBOmw9IYtpXyup_1GNqMC0gCLcBGAs/s1600/image031.png)

**Kỹ thuật lọc trung bình**

Giả sử ta có ảnh I, điểm ảnh P, cửa sổ W(P) và ngưỡng.

Khi đó kỹ thuật lọc trung bình phụ thuộc không gian bao gồm các bước cơ bản sau:

* Bước 1: Tìm trung bình {I(q)| q ∈ W(P)} → AV(P)
* Bước 2: Gán giá trị

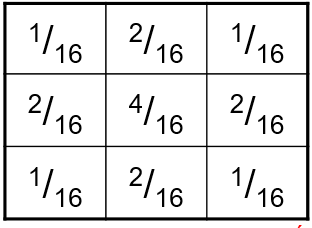
[](https://3.bp.blogspot.com/-QtrGRZWxYy0/W8BvgsIUlsI/AAAAAAAABmY/9dz5is1zAV8g7LaFssHRdLJTXAcyR36oQCLcBGAs/s1600/image032.png)

Giá trị 16 sau phép lọc trung bình có giá trị 3, các giá trị còn lại giữ nguyên sau phép lọc

1. **Lọc trung bình có trọng số**

Cho phép các pixel trong lận cận có các trọng số khác nhau trong hàm trung bình

Pixel gần với pixel trung tâm quan trọng hơn.



Hình 1.12 Bộ lọc trung bình có trọng số

Giả sử ta có ảnh I, điểm ảnh P, cửa sổ W(P), ngưỡng θ và số k. Khi đó, lọc trung bình theo k giá trị gần nhất bao gồm các bước sau:

* Bước 1: Tìm K giá trị gần nhất {I(q) ⏐q ∈ W(p)} → {k ~ giá trị gần I(P) nhất}
* Bước 2: Tính trung bình {k ~ giá trị gần I(P) nhất} → AVk(P)
* Bước 3: Gán giá trị



Nếu k lớn hơn kích thước cửa sổ thì kỹ thuật chính là kỹ thuật lọc trung bình, nếu k là 1 thì ảnh kết quả không thay đổi. Vậy nên chất lượng của kỹ thuật phụ thuộc vào số phân tử lựa chọn k

**Lọc trung vị**

**Định nghĩa (Trung vị)**

Cho dãy x1; x2...; xn đơn điệu tăng (giảm). Khi đó trung vị của dãy ký hiệu là Med({xn}), được định nghĩa:

* Nếu n lẻ :
* Nếu n chẵn: hoặc

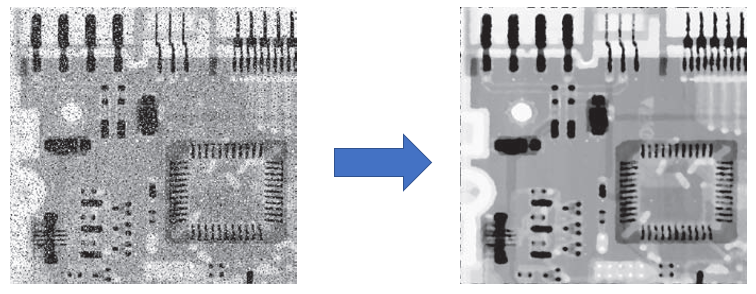
**Kỹ thuật lọc trung vị**

Giả sử ta có ảnh I ngưỡng θ cửa sổ W(P) và điểm ảnh P

Khi đó kỹ thuật lọc trung vị phụ thuộc không gian bao gồm các bước cơ bản sau:

* Bước 1: Tìm trung vị {I(q)| q ∈ W(P)} → Med (P)
* Bước 2: Gán giá trị





Hình 1.13 Lọc trung vị 7x7

1. **Bộ lọc Gauss**

Bộ lọc Gaussian:

****

Trong đó:

K : hằng số ;

σ : hằng số lệch chuẩn

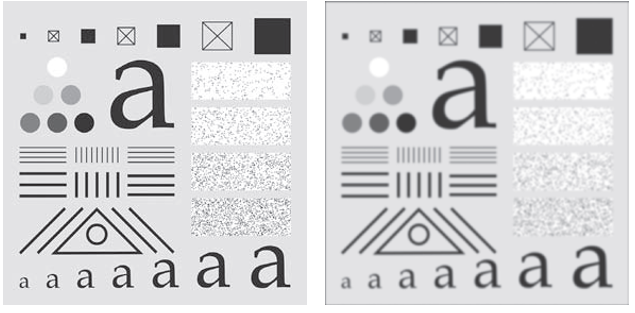
g(x,y) : là ảnh sau khi lọc;

f(x,y) : là ảnh đầu vào;

w(s,t): mặt nạ lọc

a = (m-1)/2 và b = (n-1)/2;

m x n : là kích thước bộ lọc



Hình 1.14 Ảnh lọc Gaussian kích thước 21 x 21 khi K = 1; σ =3.5

#### Kỹ thuật lọc sắc nét ảnh

1. **Lọc đạo hàm bậc 1**

Gradient là một vectơ có các thành phần biểu thị tốc độ thay đổi mức xám của điểm ảnh (theo hai hướng *x,y* đối với ảnh 2 chiều) tức là:



Ta có:



Trong đó *dx, dy* là khoảng cách giữa 2 điểm kế cận theo hướng x,y tương ứng (thực tế chọn *dx=dy*=1)

Nếu áp dụng Gradient vào xử lý ảnh, việc tính toán sẽ rất phức tạp.

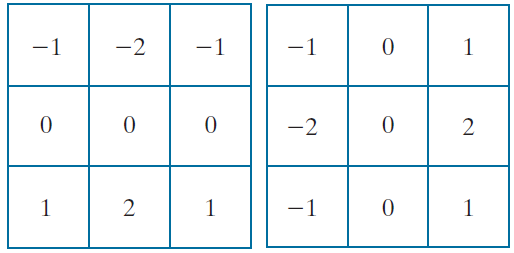
Để đơn giản mà không mất tính chất của phương pháp Gradient, người ta sử dụng kĩ thuật Gradient dùng cặp mặt nạ *H­1,H2* trực giao. Nếu định nghĩa Gx, Gy tương ứng là Gradient theo hai hướng x,y khi đó ta có vector Gradient của một ảnh f(x,y) là:

ta có 

* Bộ lọc Sobel

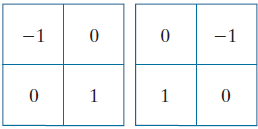
Tổng các phần tử quanh điểm trung tâm bằng 0

Để lọc ảnh, chúng ta sử dụng cả 2 toán tử và kết hợp (cộng) kết quả lại với nhau



Hình 1.15 Bộ loc Sobel

* Bộ lọc Robert cross gradient



Hình 1.16 Bộ lọc Robert cross gradient

1. **Lọc đạo hàm bậc 2**

* Bộ lọc Laplacian

Laplacian được định nghĩa như sau:



Trong đó đạo hàm riêng bậc 2 theo phương x được định nghĩa như sau:



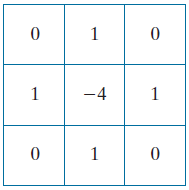
Và theo phương y như sau:



Do vậy, Laplacian được viết lại như sau:

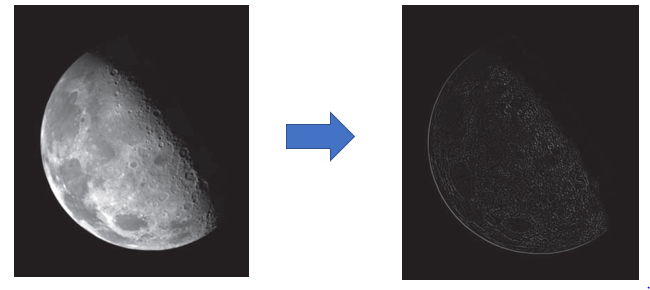


Và chúng ta dễ dàng xây dựng bộ lọc như sau:



Hình 1.17 Bộ lọc Laplacian

Áp dụng Laplacian vào một ảnh, chúng ta nhận được một ảnh làm nỗi biên và các đường nét không liên tục khác



Hình 1.18 Ví dụ ảnh sử dụng bộ lọc Laplacian

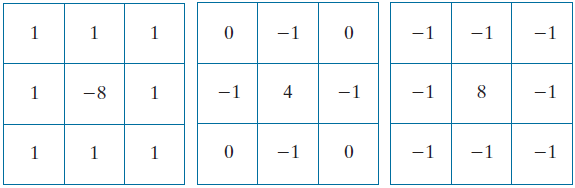
Kết quả của lọc Laplacian không phải là một ảnh cải thiện

Chúng ta phải thực hiện thêm thao tác để có được ảnh cuối cùng

Trừ ảnh ban đầu cho ảnh Laplacian để được ảnh sau cùng ảnh cải thiện và sắc nét



Các biến thể khác nhau của Laplacian:



Hình 1.19 Các biến thể khác nhau của Laplacian

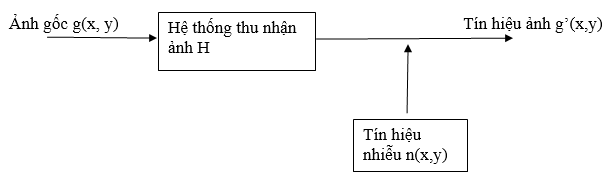
## Khôi phục ảnh

### Nhiễu và mô hình nhiễu

Ảnh được coi là một miền đồng nhất về mức xám, tức là các điểm ảnh lân cận có sự biến đổi liên tục về mức xám. Như vậy sau quá trình số hoá thì trong mỗi cửa sổ đang xét các điểm ảnh đều có giá trị gần bằng như nhau. Thực tế quan sát có những điểm ảnh có giá trị khác hơn nhiều so với các điểm ảnh xung quanh. Đó chính là nhiễu. Như vậy, nhiễu trong ảnh số được xem như là sự dịch chuyển đột ngột của tín hiệu ảnh trên một khoảng cách nhỏ.

* **Mô hình liên tục**

Hệ thống thu nhận ảnh chuyển các hình ảnh thực của môi trường xung quanh g(x,y) thành dạng tín hiệu ảnh g’(x,y). Tuy nhiên trong quá trình chuyển đổi có nhiều yếu tố tác động tạo thành nhiễu. Do đó tín hiệu g’(x,y) có thể chứa các thành phần nhiễu trong đó. Quá trình thu nhận ảnh có nhiễu có thể mô tả một cách trực quan như sau:



Hình 1.20 Mô hình nhiễu

* **Mô hình rời rạc:**

Trên cơ sở mô hình nhiễu liên tục, ta có thể xây dựng một mô hình nhiễu rời rạc tương ứng với ảnh số. Khi đó g(x,y) sẽ chuyển thành ảnh rời rạc g[m,n], ảnh liên tục g’(x,y) sẽ chuyển thành ma trận điểm ảnh g’[m,n] và nhiễu cũng phân bố rời rạc tại các điểm ảnh η[m,n]. Giả sử H là hàm tuyến tính bất biến trong phạm vi M×N (kích thước ảnh) thì ta có:



Đối với xử lý số ảnh thì ta chỉ sử dụng mô hình nhiễu rời rạc.

### Các loại nhiễu

Các tín hiệu nhiễu thường được chia thành các loại chính như sau:

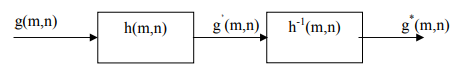
* Nhiễu do thiết bị thu nhận ảnh: là loại nhiễu gây ra do giới hạn nhiễu xạ và quang sai của thấu kính, nhiễu do bộ phận cảm quang, ảnh mờ nhòe do ống kính, nhiễu do rung động thiết bị trong quá trình thu nhận.
* Nhiễu nhẫu nhiên độc lập: là các loại nhiễu gây ra do ảnh hưởng của môi trường xung quanh, do ảnh hưởng của khí quyển.
* Nhiễu do vật quan sát: là nhiễu gây ra do bề mặt của bản thân vật có độ nhám gồ ghề. Chính nhiễu này gây hiện tượng tán xạ của các tia đơn sắc và sinh ra hiện tượng nhiễu lốm đốm.

Thường người ta xấp xỉ các loại nhiễu bằng các quá trình tuyến tính bất biến vì có nhiều công cụ tuyến tính có thể giải quyết vấn đề khôi phục ảnh hơn là các công cụ phi tuyến. Việc xử lý nhiễu bằng cách xấp xỉ tuyến tính cũng giúp cho công việc dễ dàng hơn trong trường hợp dùng cách biến đổi phi tuyến.

### Các kỹ thuật lọc nhiễu

1. Lọc đảo (inverse filter):

Nguyên lý của lọc đảo là sử dụng hàm ngược của đáp ứng xung h[m,n] để khôi phục lại một ảnh xấp xỉ ảnh nguyên gốc g[m,n] từ ảnh g’[m,n] đã biết nguyên lý này được biễu diễn mô tả theo sơ đồ sau:



Hình 1.21 Lọc đảo khôi phục ảnh nguyên

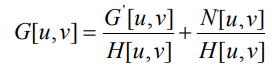
Như vậy ảnh khôi phục g\*[m,n] sẽ được tính theo công thức:

g\*[m,n]= g’[m,n]\* h-1 [m,n] (3.4)

Trong đó h-1 [m,n] chính là hàm của bộ lọc đảo. Vì H-1 (H(x)) = x nên ta có giá trị đầu ra là g\*[m,n] cũng bằng giá trị đầu vào g[m,n]. Như vậy ta đã khôi phục được ảnh g[m,n] nhờ dùng hàm ngược của đáp ứng xung h[m,n]. Nếu dùng biến đổi Fourier ta có:

G[u,v]=G’[u,v] HT[u,v]= G’[u,v] /HT[u,v]

Qua đó ta thấy, đáp ứng tần số của bộ lọc đảo là nghịch đảo của đáp ứng tần số của hệ thu nhận ảnh. Nếu đánh giá được mức nhiễu ta có thể xấp xỉ gần hơn với ảnh nguyên gốc:

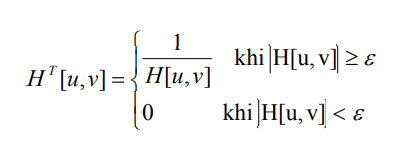


với N[u,v] là nhiễu ước lượng

Nếu H[u,v] bằng 0 hoặc khá nhỏ thì hệ thống khôi phục sẽ không ổn định (hàm HT [u,v] không xác định). Đây chính là nhược điểm của phương pháp lọc đảo. Tuy bộ lọc đảo có khả năng ngăn nhiễu do hệ thống nhận ảnh gây ra khá tốt nhưng việc thiết kế bộ lọc này lại là khá phức tạp.

1. Lọc giả đảo (Pseudoinverse filter):

Kỹ thuật lọc này khắc phục được nhược điểm của kỹ thuật lọc đảo là làm cho hàm HT [u,v] luôn xác định:



Với ε là một giá trị cho trước. Trong trường hợp ảnh nguyên gốc g[m,n] chuyển động tịnh tiến theo phương x và y sinh ra hiện tượng nhoè ảnh thì HT [u,v] sẽ được xác định theo các thông số chuyển động, và lọc giả đảo có khả năng khôi phục được ảnh nhoè này.

1. Lọc nhiễu lốm đốm:

Ta đã biết nhiễu lốm đốm gây ra do tính chất gồ ghề của bề mặt vật thể gây ra hiện tượng tán xạ các tia đơn sắc. Phần này ta sẽ nghiên cứu một kỹ thuật lọc nhiễu lốm đốm là kỹ thuật trung bình thống kê sử dụng bộ lọc đồng cầu.

Kỹ thuật lọc nhiễu trung bình thống kê tiến hành thống kê các cường độ sáng của đối tượng ảnh bị nhiễu lốm đốm bằng N lần thu nhận độc lập và lấy trung bình các cường độ đó.

Ta giả thiết hệ thống chỉ có nhiễu lốm đốm và nhiễu này có thể coi như là tổng vô số hạn các pha độc lập và đồng nhất. Ta có thể biễu diễn nhiễu lốm đốm như sau:

a[m,n]=aR[m,n]+j.aL[m,n]

Với aR và aL là các biến ngẫu nhiên độc lập phân bố theo Gaussian, ứng với mỗi toạ độ [m,n] có trung bình bằng 0 và phương sai σ2. Ta có cường độ S

S[m,n] = | a[m,n] | = ar2 + al2

Như vậy ảnh thu được trong lần nhận thứ i (1,...,N) sẽ là:

gi’[m,n]=g[m,n].si[m,n]

Lọc đồng cầu (Homomorphic filter):

Nếu lấy logarit 2 vế của biểu thức ta thu được:

log g’N[m,n]= log g[m,n]+ log sN[m,n]

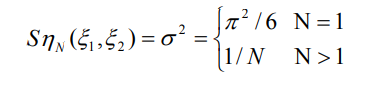
Đặt wN[m,n]= log g’N[m,n], z[m,n]= log g[m,n],ηN[m,n]= log sN[m,n]

Ta có mô hình quan sát có nhiễu lốm đốm như sau:

wN[m,n]= z[m,n] +ηN[m,n] (3.12)

Từ công thức 3.12 ta thấy có thể tìm ra z[m,n] từ wN[m,n] và ηN[m,n]. Như vậy là có thể tách nhiễu lốm đốm và khôi phục lại ảnh ban đầu.

Trong công thức 3.12 thì ηN[m,n] là nhiễu trắng dừng. Với N≥2 thì ηN[m,n] có thể mô tả gắn với nhiễu ngẫu nhiên Gauusian với mật độ phổ được định nghĩa như sau:



Toàn bộ lý thuyết trên được tham khảo tại tài liệu của môn xử lý ảnh nằm ở tài liệu Xử lý ảnh

## Không gian màu

### Không gian màu HSV

HSV hay còn gọi là không gian màu HSB là một không gian màu dựa trên 3 thông số chính của không gian màu:

* H viết tắt của từ HUE có nghĩa là vùng màu
* S Viết tắt của từ SATURATION có nghĩa là độ bảo hòa màu.
* V viết tắt của chữ VALUE có nghĩ là giá trị hay độ sáng của màu sắc

#### Cách sử dụng không gian màu HSV

Bánh xe màu HSV đôi khi được mô tả dưới dạng hình nón hoặc hình trụ, nhưng luôn có ba thành phần sau:

1. Hue

Hue là phần màu của mô hình màu và được biểu thị dưới dạng một số từ 0 đến 360 độ:



Hình 1.22 Thành phần Hue

1. Saturation

Độ bão hòa là lượng màu xám trong màu, từ 0 đến 100 phần trăm. Một hiệu ứng mờ nhạt có thể có được từ việc giảm độ bão hòa về không để giới thiệu nhiều màu xám hơn.

Tuy nhiên, độ bão hòa đôi khi được xem trên phạm vi từ 0-1, trong đó 0 là màu xám và 1 là màu chính.

1. Value

Giá trị hoạt động kết hợp với độ bão hòa và mô tả độ sáng hoặc cường độ của màu sắc, từ 0-100 phần trăm, trong đó 0 là hoàn toàn đen và 100 là sáng nhất và cho thấy màu sắc nhất.

#### Ứng dụng của không gian màu HSV

Không gian màu HSV được sử dụng khi chọn màu cho sơn hoặc mực vì HSV thể hiện rõ hơn cách mọi người liên quan đến màu sắc hơn không gian màu RGB.

Bánh xe màu HSV cũng được sử dụng để tạo ra đồ họa chất lượng cao. Mặc dù ít được biết đến hơn so với người anh em họ RGB và CMYK, phương pháp HSV có sẵn trong nhiều chương trình phần mềm chỉnh sửa hình ảnh cao cấp.

Chọn màu HSV bắt đầu bằng cách chọn một trong các màu có sẵn, đó là cách hầu hết mọi người liên quan đến màu sắc, sau đó điều chỉnh giá trị độ sáng và độ bóng.

HSV thường được dử dụng trong chỉnh sửa ảnh

### Không gian màu RBG

RGB là không gian màu phổ biến dùng trong máy tính, máy ảnh, điện thoại và nhiều thiết bị kĩ thuật số khác nhau. Không gian màu này khá gần với cách mắt người tổng hợp màu sắc. Nguyên lý cơ bản là sử dụng 3 màu sắc cơ bản R (red - đỏ), G (green - xanh lục) và B (blue - xanh lam) để biểu diễn tất cả các màu sắc.

Thông thường, trong mô hình 24 bit mỗi kênh màu sẽ sử dụng 8bit để biểu diễn, tức là giá trị R, G, B nằm trong khoảng 0 - 255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu. Với mô hình màu 24bit thì số màu tối đa có thể tạo ra là 255 x 255 x 255 = 16581375 màu. Một điểm cân lưu ý là với các thư viện đọc ảnh và hiển thị ảnh như matplotlib, Pillow thì các ảnh được đọc theo RGB tuy nhiên **Opencv đọc ảnh theo các kênh BGR**.

Bên cạnh hệ màu RGB thì ta sẽ nghe đến RGBA, thực ra đây là một hệ màu được lấy căn bản từ hệ màu RGB tuy nhiên có thêm một kênh alpha (α). Kênh α được sử dụng như là kênh mờ, nếu một pixel có giá trị 0% trong kênh α của nó thì nó hoàn toàn trong suốt, trong khi giá trị 100% sẽ khiến cho điểm đó bị mờ đục. Điều này được ứng dung rất nhiều trong việc ghép các ảnh lại và mang một độ chân thực nhất định

### Không gian màu CMYK

CMYK là từ viết tắt tiếng Anh của cơ chế hệ màu trừ, thường được sử dụng trong in ấn. Nó bao gồm các màu sau:

* C = Cyan (xanh)
* M = Magenta (hồng)
* Y = Yellow (vàng)
* K = Black (Đen) (sở dĩ dùng từ K để chỉ màu đen vì ký tự B đã được dùng để chỉ màu Blue, ngoài ra K còn có nghĩa là Key, mang ý chỉ cái gì đó là chủ yếu, là then chốt)

#### Nguyên lý hoạt động

Nguyên lý làm việc chính của hệ CMYK là hấp thụ ánh sáng. Màu mà ta nhìn thấy là từ phần của ánh sáng không bị hấp thụ, hay nói cách khác, chúng hoạt động trên cơ chế những vật không tự phát ra ánh sáng mà chỉ phản xạ ánh sáng từ các nguồn khác chiếu tới.

Do đó thay vì thêm độ sáng để có những màu sắc khác nhau, CMYK sẽ loại trừ ánh sáng đi từ ánh sáng gốc là màu trắng để tạo ra các màu sắc khác. 3 màu Cyan, Magenta và Yellow khi kết hợp sẽ tạo ra một màu đen.

Màu CMYK thường được sử dụng khi thiết kế phục vụ cho mục đích in ấn các thiết kế như poster, brochure, name card, catalogue, sách hoặc tạp chí,…

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Mô tả yêu cầu

Yêu cầu chức năng

Chương trình xử lý hình ảnh với các chức năng như sau:

* Tạo ra chương trình xử lý ảnh với các chức năng chính như:
* Làm sắc nét
* Làm mịn
* Tăng giảm độ sáng
* Tăng tương phản
* Tăng kích thước ảnh
* Crop ảnh

Yêu cầu phi chức năng

* Thanh menu thật đơn giản.
* Font chữ đơn giản, dễ nhìn, màu sắc hài hòa.

## Ngôn ngữ lập trình và các thư viện hỗ trợ

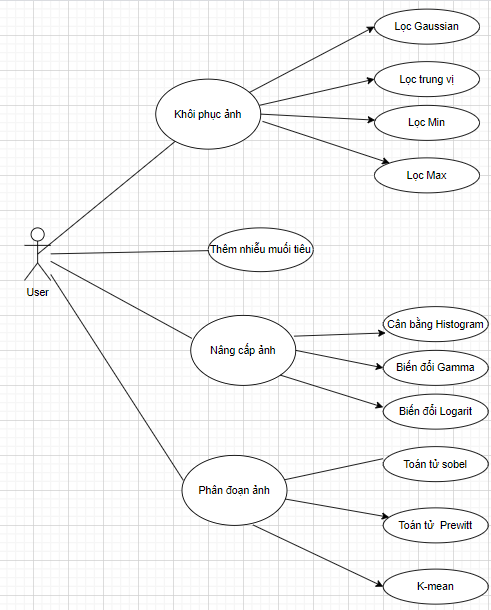
Trong phạm vi đồ án này, chúng em sử dụng ngôn ngữ lập trình Python phiên bản 3.7.

Các thư viện hỗ trợ:

* Opencv (Open Computer Vision)**:** là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, IOS.
* Numpy: là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó.
* Tensorflow: là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.
* Keras:là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một library được phát triển vào năm 205 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning. Phần backend của keras thường là Tensorflow
* Matplotlib: Sử dụng để vẽ biểu đồ histogram.
* PyQt5: Giao diện chính của chương trình được xây dựng bằng thư viện PyQt5 của python. PyQt5 là một thư viện của Python cho phép tạo GUI với framework Qt5, cho phép xây dựn giao diện desktop app trên python, có thể tinh chỉnh các thuộc tính bằng stylesheet tương tự như css ở trên web.

## Phân tích

### Biểu đồ Usecase



Hình 2.1 Biểu đồ usecase

### Đặc tả usecase

Table 2‑1 Đặc tả usecase mở ảnh

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử dụng mở ảnh | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người mở ảnh | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng mở ảnh |  |
| 2 |  | Đưa ra giao diện chọn ảnh |
| 3 | Chọn ảnh cần xử lý |  |
| 4 | Nhấn nút “Open” |  |
| 5 |  | Hệ thống xử lý mở ảnh, hiển thị ảnh lên giao diện. |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện**  **sau** |  | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

Table 2‑2 Đặc tả usecase zoom ảnh

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử Zoom ảnh | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người Zoom ảnh | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng Zoom ảnh |  |
| 2 |  | Đưa ra giao diện chọn ảnh sau khi đã xử lý Zoom |
|  |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

Table 2‑3 Đặc tả usecase thêm nhiễu

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử thêm nhiễu | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người thêm nhiễu vào ảnh | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng thêm nhiễu Filter->Noise->Add Noise |  |
| 2 |  | Đưa ra giao diện chọn ảnh sau khi đã xử lý thêm nhiễu |
|  |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

Table 2‑4 Đặc tả usecase lọc sobel

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử lọc sobel | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người lọc nhiễu | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng Filter-Sharpen->Sobel |  |
| 2 |  | Đưa ra giao diện chọn ảnh sau khi đã xử lý lọc ảnh |
|  |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

Table 2‑5 Đặc tả usecase phân vùng ảnh

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử phân vùng ảnh | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người phân vùng ảnh | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng phân vùng ảnh |  |
| 2 | Chọn số màu phân vùng |  |
| 3 |  | Đưa ra giao diện chọn ảnh sau khi đã xử lý phân vùng ảnh |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

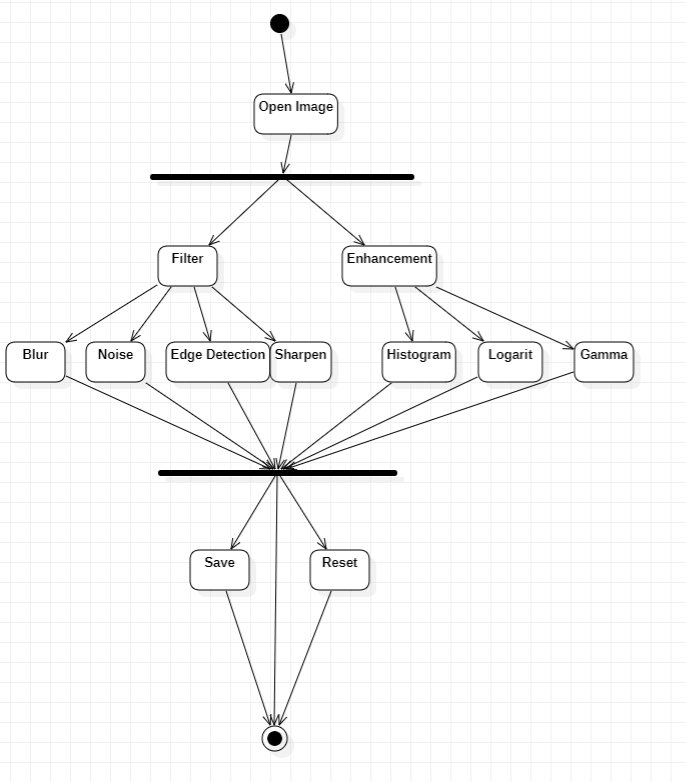
Table 2‑6 Đặc tả usecase cần bằng Histogram

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử cân bằng Histogram | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người cân bằng histogram | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng Enhancement ->Histogram equal |  |
| 2 |  | Đưa ra giao diện chọn ảnh sau khi đã xử lý histogram |
|  |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

Table 2‑7 Đặc tả usecase lọc gaussian

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên ca sử**  **dụng** | Ca sử lọc Gaussian | | |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Cho phép người lọc ảnh Gaussian | | |
| **Luồng sự kiện** |  | **Tác nhân** | **Hệ thống phản hồi** |
| 1 | Chọn chức năng Filter->Blur Gaussian |  |
| 2 |  | Đưa ra giao diện chọn số lọc Gaussian |
| 3 | Kéo chọn số cho bộ lọc. Sau đó chọn Oke |  |
| 4 |  | Hiển thị ảnh đã xử lí lọc Gaussian |
|  |
| **Luồng thay thế và ngoại lệ** |
| **Điều kiện trước** | * Người dùng đã mở ảnh | | |
| **Điều kiện thoát** | * Khi người dùng chọn thoát | | |

### Biểu đồ hoạt động



Hình 2.2 Biểu đồ hoạt động

# XÂY DỰNG CHƯƠNG TRÌNH XỬ LÝ ẢNH

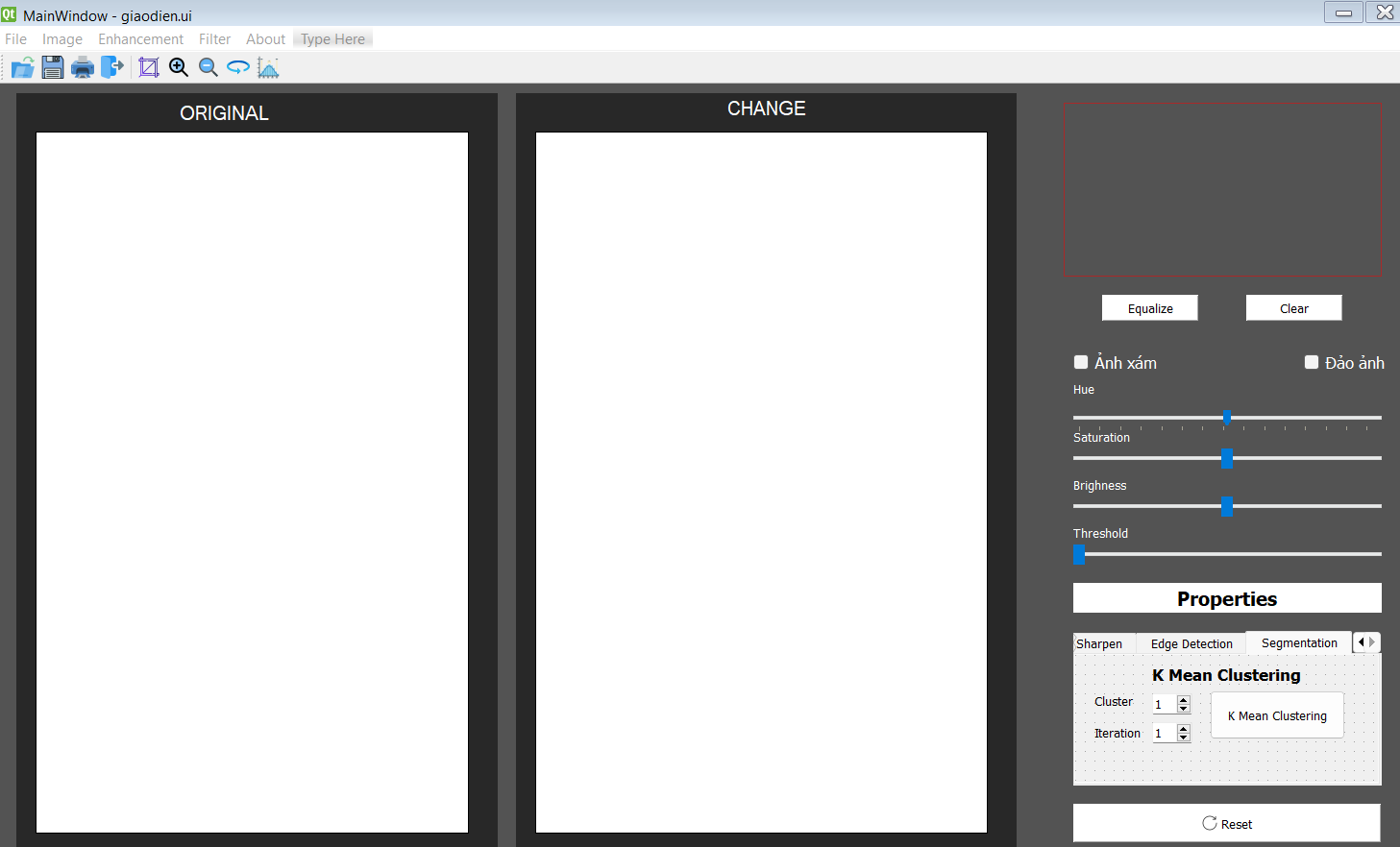
Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lƣợng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

## Xây dựng giao diện

Giao diện chính của chương trình được xây dựng bằng thư viện PyQt5 của python

Giao diện chương trình bào gồm 3 phần:

* Phần hiển thị ảnh gốc và ảnh qua chỉnh sửa
* Phía bên phải là phần hiển thị các chức năng: bao gồm các Slider, các Check box, các Radio button, Combobox và biểu đồ histogram nằm phía trên
* Phía trên phần hiển thị hình ảnh chứa các nút công cụ như mở hình ảnh, lưu hình ảnh, ẩn/ hiện biểu đồ histogram, cắt ảnh , vẽ



Hình 3.1 Giao diện phần mềm

## Xây dựng chức năng

### Chức năng đảo ngược mức sáng

Chức năng này được thực hiện với một dòng lệnh của thư viện numpy

image = 255 – image

### Chức năng phân ngưỡng

Chức năng này có sẵn trong thư viện OpenCV với hàm cv2.threshold, trong đó tham số thứ nhất là ảnh đầu vào, tham số thứ 2 là ngưỡng, tham số thứ 3 là giá trị mức sáng sao nhất, tham số thứ 4 là hàm phân ngưỡng, thường dung là hàm THRESH\_BINARY

image = cv2.threshold(image,threshsold\_value,255,cv2.THRESH\_BINARY)

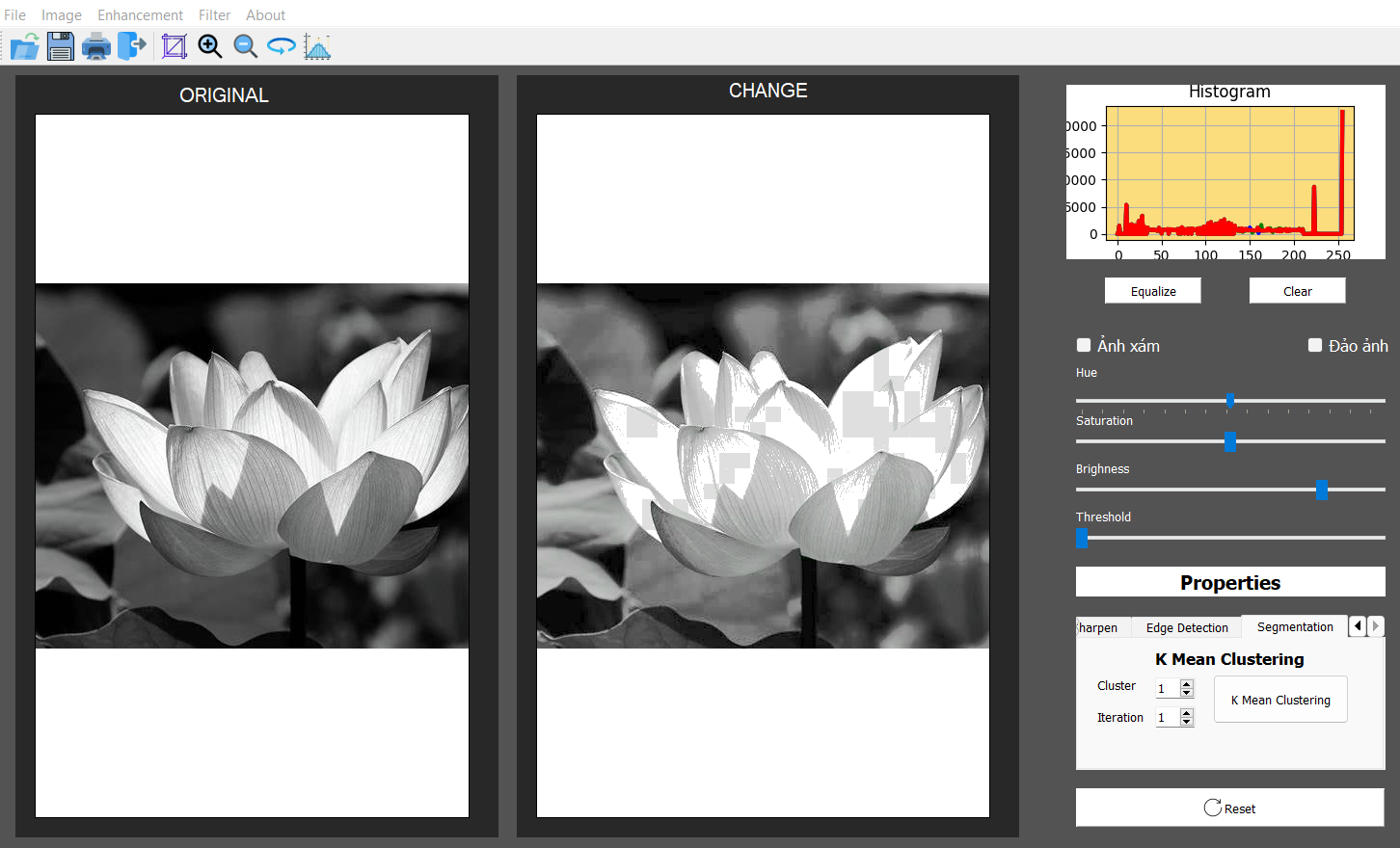
### Chức năng điều chỉnh độ tương phản

Để biến đổi độ tương phản ta sử dụng công thức:

g(x) = α f(x) + β với α >0

Nếu α < 1 và β > 0 thì làm giảm tương phản của ảnh

Nếu α > 1 và β < 0 thì làm tăng tương phản của ảnh



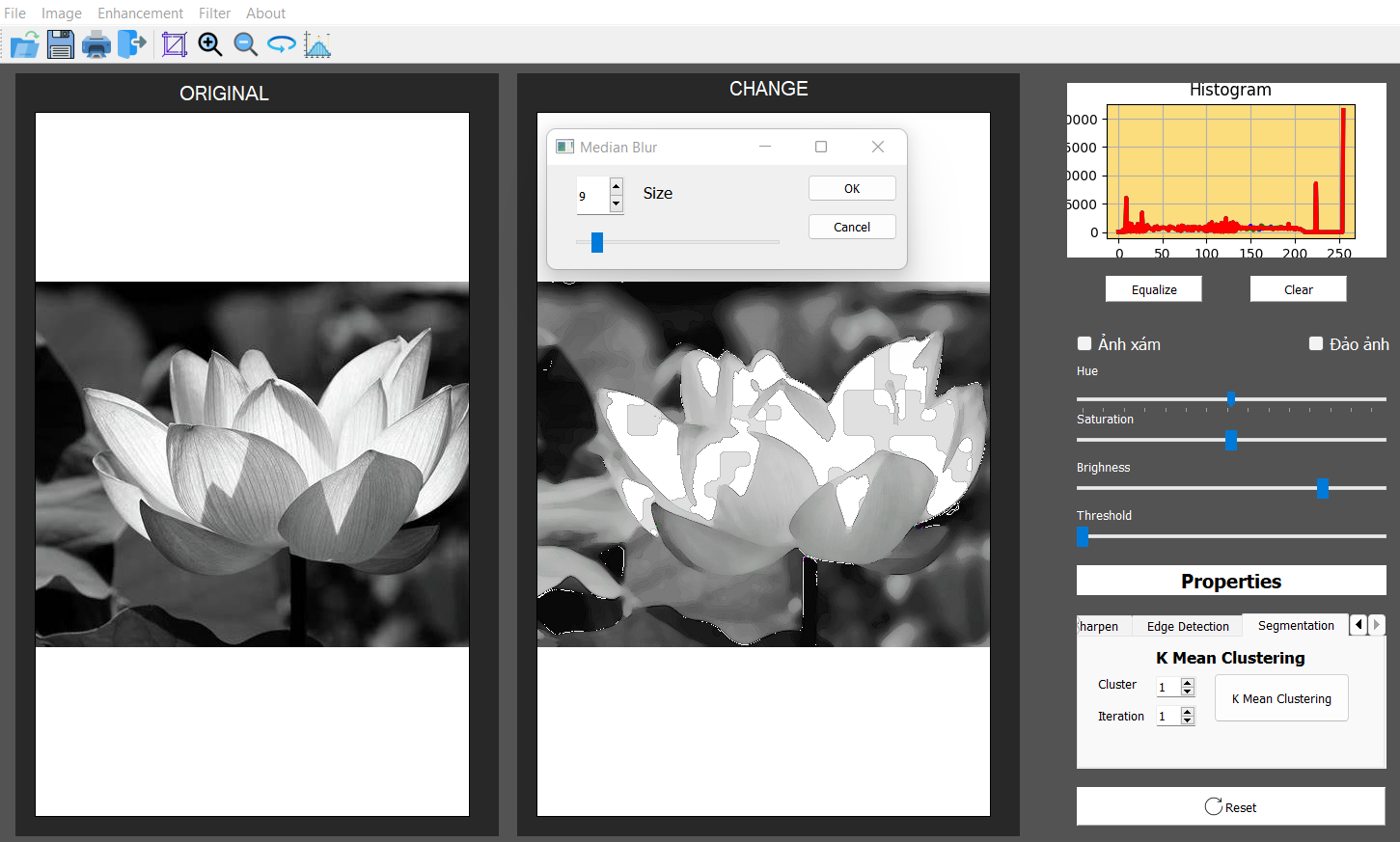
Hình 3.2 Điều chỉnh độ tương phản

### Chức năng lọc ảnh

Thư viện OpenCV có hàm filter2D cho phép thực hiện lọc trong miền không gian rất nhanh chóng, vì vậy, việc còn lại là chỉ cần đưa mặt nạ lọc, ảnh đầu vào nữa là được.

Cài đặt: mask là mặt nạ lọc, 1 mảng numpy

image = cv2.filter2D(image, -1, mask)



Hình 3.3 Lọc Gaussian

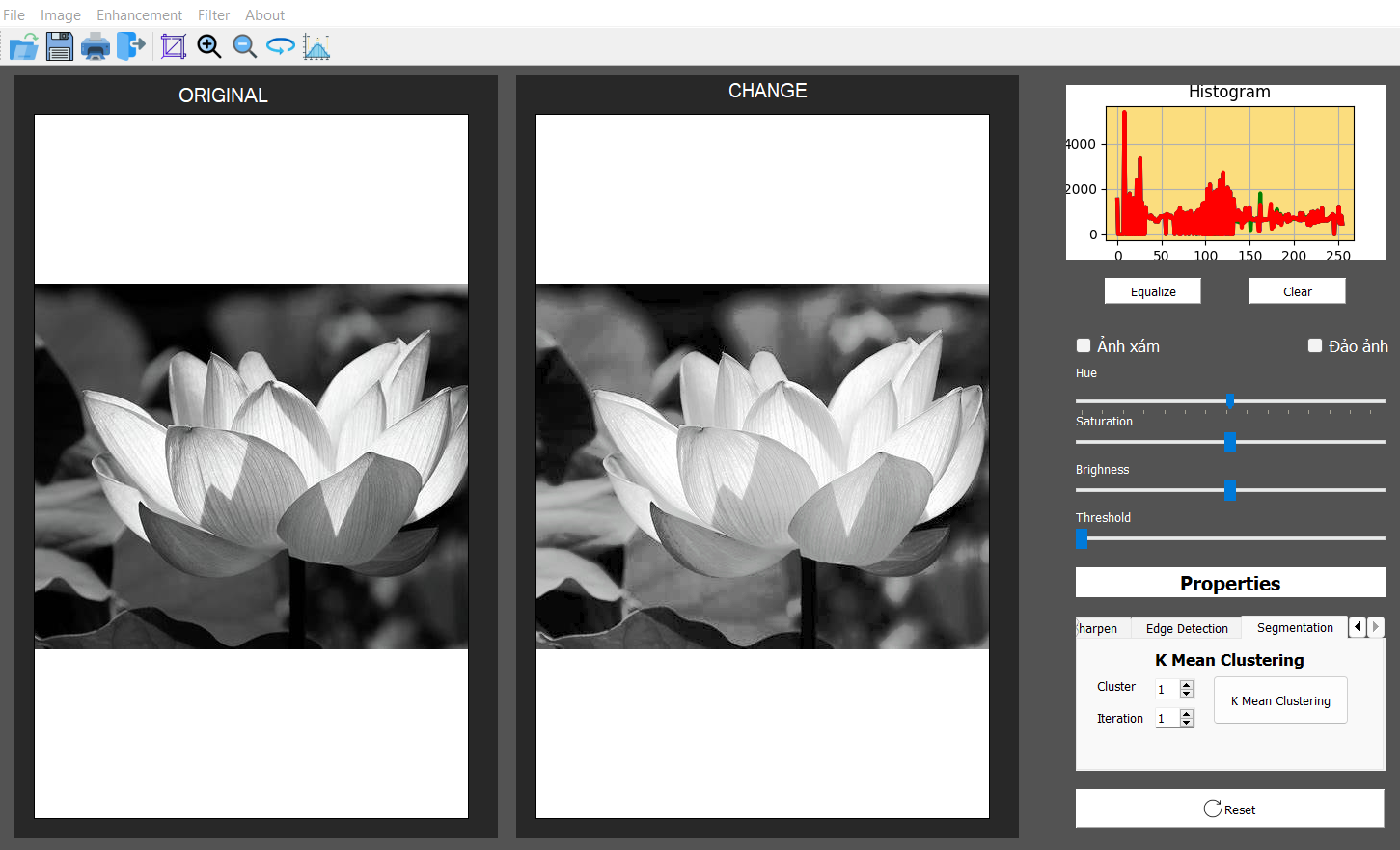
### Chức năng cân bằng histogram

Thống kê số lượng pixel cho từng mức sáng, ta được histogram H(i)*H*(*i*)

**B2**: Tính "hàm tích lũy" Z*Z* cho từng mức sáng theo công thức:

Z(i) = \sum\_{j=0}^{i} H(i)*Z*(*i*)=*j*=0∑*i*​*H*(*i*)

Trong đó Z(i)*Z*(*i*) chính là tổng số pixel có giá trị \leqslant i⩽*i* . Trên hình H1, đường màu đỏ chính là đường minh họa Z(i)*Z*(*i*). Z(i)*Z*(*i*) chắc chắn là hàm đồng biến tăng.



Hình 3.4 Cân bằng Histogram

KẾT LUẬN

##### Kết quả đạt được

* Về mặt lý thuyết :
  + Hiểu được các kiến thức và kỹ thuật xử lý ảnh
  + Tìm hiểu, phân tích các mô hình thực thể trong chương trình xử lý ảnh
* Về mặt thực nghiệm :
  + Phân tích, thiết kế các chức năng của chương trình xử lý ảnh
  + Vận dụng các kiến thức lý thuyết đã nắm được để xây dựng một chương trình xử lý ảnh với các chức năng cơ bản

##### Hạn chế

* Các tính năng còn chỉ ở mức cơ bản
* Giao diện còn đơn giản

##### Hướng phát triển

* Tiếp tục cải tiến giao diện để bắt mắt và phù hợp với nhiều người dùng
* Phát triển thêm chức năng xử lý ảnh với AI cho chương trình

TÀI LIỆU THAM KHẢO

##### 1. Không gian màu HSV:

<https://inanlubi.vn/tin-tuc/khong-gian-mau-sac-hsv>

##### 2. Các hệ màu trong xử lý ảnh:

<https://kipalog.com/posts/Cac-he-mau-trong-xu-ly-anh>

##### 3. Tham khảo cài đặt mô hình unet:

<https://phamdinhkhanh.github.io/2020/06/20/Unet.html>

<https://www.kaggle.com/arunkothari/image-segmentaion-unet-99-acc>

##### 4. *Xử lý ảnh* - PGS.TS .Đỗ Năng Toàn - HVCNBCVT -2013

<https://www.academia.edu/15561188/B%C3%A0i_gi%E1%BA%A3ng_x%E1%BB%AD_l%C3%BD_%E1%BA%A3nh>