

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**

---  ---



ĐỒ ÁN TỐT NGHIỆP

NGHIÊN CỨU THUẬT TOÁN MÃ HÓA BẢO VỆ HÌNH ẢNH Y TẾ VÀ THỰC NGHIỆM

SINH VIÊN THỰC HIỆN : ĐINH CÔNG LÂM
MÃ SINH VIÊN : 1451020133
KHOA : CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



ĐINH CÔNG LÂM

**NGHIÊN CỨU THUẬT TOÁN MÃ HÓA
BẢO VỆ HÌNH ẢNH Y TẾ VÀ THỰC
NGHIỆM**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SỐ : 74.80.201**

NGƯỜI HƯỚNG DẪN: TS. BÙI HẢI PHONG

HÀ NỘI – 2024

LỜI CAM ĐOAN

Kính thưa Quý thầy cô giáo.

Em là: Đinh Công Lâm

Mã số sinh viên: 1451020133

Là sinh viên Trường Đại học Đại Nam Niên khóa : 2020 – 2024

Em xin cam đoan đề tài “ Nghiên cứu thuật toán mã hóa bảo vệ hình ảnh y tế và thực nghiệm” là công trình nghiên cứu độc lập của bản thân, không sao chép của người khác. Mọi số liệu trong đề án này là hoàn toàn có thật và được lấy từ những nguồn đáng tin cậy. Nếu sai, em xin hoàn toàn chịu trách nhiệm và kỷ luật từ phía nhà trường.

Hà Nội, ngày 25 tháng 05 năm 2024.

Sinh viên thực hiện

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành nhất đến các thầy cô giáo tại trường, những người đã truyền đạt kiến thức và kinh nghiệm quý báu, giúp em hoàn thành đồ án này.

Em xin gửi lời cảm ơn đặc biệt đến giáo viên hướng dẫn thầy **TS. Bùi Hải Phong**, người đã hỗ trợ em không chỉ về mặt chuyên môn mà còn về tinh thần trong suốt quá trình thực hiện đồ án.

Em hiểu rằng, mặc dù đã cố gắng hết sức, đồ án của em vẫn còn nhiều hạn chế. Em rất mong nhận được sự góp ý và phê bình của các thầy cô và các bạn để đồ án của em có thể hoàn thiện hơn trong tương lai.

Xin chân thành cảm ơn!

LỜI NÓI ĐẦU

Với sự phát triển nhanh chóng của công nghệ thiết bị y tế, việc chẩn đoán các bệnh khác nhau bằng hình ảnh y tế đã trở nên phổ biến. Hình ảnh y tế được truyền qua các mạng khác nhau, do đó, việc bảo mật những hình ảnh này đã trở thành một chủ đề thiết yếu trong những năm gần đây. Việc truyền hình ảnh y tế một cách an toàn đòi hỏi tính bảo mật, tính toàn vẹn và tính xác thực. Việc sử dụng trái phép những hình ảnh như vậy có thể dẫn đến mất quyền riêng tư đối với dữ liệu của bệnh nhân. Hơn nữa, khi những hình ảnh này có bất kỳ sự thay đổi nhỏ nào có thể dẫn đến chẩn đoán không chính xác và có thể đe dọa đến tính mạng của bệnh nhân.

Nói chung, có thể bảo mật hình ảnh kỹ thuật số bằng cách sử dụng kỹ thuật giấu ảnh (steganography), thủy vân (watermarking) và mã hóa hình ảnh. Mã hóa là phương pháp đơn giản và hiệu quả nhất để đảm bảo an toàn cho hình ảnh y tế thông qua việc chuyển đổi hình ảnh rõ ban đầu thành hình ảnh không thể đọc được bằng khóa bí mật. Không có khóa bí mật đó, không ai có thể khôi phục lại hình ảnh ban đầu. Mã hóa hình ảnh phụ thuộc vào hai yếu tố chính: tính hỗn loạn (confusion) và tính khuếch tán (diffusion).

Các thuật toán đưa ra để bảo mật hình ảnh y tế đều có thể bị tấn công. Mối tương quan chặt chẽ giữa các pixel lân cận đặc trưng cho hình ảnh y tế, do đó việc loại bỏ mối tương quan này đòi hỏi một kỹ thuật hoán vị (xáo trộn) với mức bảo mật cao hơn.

Trong đồ án tốt nghiệp này, thuật toán dựa trên bốn phần chính: tách hình ảnh, xáo trộn hình ảnh, tạo khóa và khuếch tán. Đầu tiên, hình ảnh được chia thành các khối và khối con bằng kỹ thuật tách hình ảnh. Thứ hai, sự sắp xếp của các pixel được thay đổi trong các khối và khối con bằng cách sử dụng mẫu zigzag, xoay một góc 90 độ và hoán vị ngẫu nhiên giữa các khối. Thứ ba, một khóa được tạo ra từ logistic map, trong đó điều kiện ban đầu của logistic map phụ thuộc vào hình ảnh gốc. Cuối cùng, các giá trị pixel của ảnh được thay đổi bằng khóa bí mật.

Báo cáo gồm ba phần chính, được tổ chức như sau:

1. Chương 1: “Tổng quan về ảnh số và một số phương pháp mã hóa ảnh” cung cấp một cái nhìn tổng quan về ảnh số và công việc được thực hiện trong lĩnh vực mã hóa hình ảnh cũng như những nội dung về xử lý ảnh.
2. Chương 2: “Thuật toán mã hóa bảo vệ hình ảnh y tế” mô tả ngắn gọn về công việc mã hóa và giải mã dựa trên thuật toán bảo vệ hình ảnh y tế.
3. Chương 3: “Tiến hành mã hóa ảnh và giải mã hình ảnh y tế” trình bày kết quả thực nghiệm và những đánh giá chủ quan của em đối với thuật toán nghiên cứu.

Sau thời gian thực hiện, các mục tiêu cơ bản được đặt ra khi thực hiện đề tài này đã đạt được. Tuy nhiên, do vốn kiến thức và kinh nghiệm còn hạn chế nên bài báo cáo còn nhiều thiếu sót. Em rất mong nhận được sự góp ý từ quý thầy cô.

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	1
1.1. Tính cấp thiết của đề tài.....	1
1.2. Lý do chọn đề tài	1
1.3. Mục đích nghiên cứu	2
1.4. Phạm vi nghiên cứu	2
CHƯƠNG 2. TỔNG QUAN VỀ ẢNH SỐ VÀ MỘT SỐ PHƯƠNG PHÁP MÃ HÓA ẢNH.....	3
2.1. Một số khái niệm về ảnh số	3
2.1.1. Khái niệm về ảnh số	3
2.1.2. Khái niệm về pixel và số bit trên pixel	5
2.1.3. Khái niệm về histogram	5
2.1.4. Mức xám.....	6
2.2. Phân loại ảnh số	6
2.2.1. Ảnh Raster	7
2.2.2. Ảnh Vector	8
2.2.3. Ảnh đen trắng	9
2.2.4. Ảnh màu	10
2.3. Các không gian màu biểu diễn ảnh số	11
2.3.1. Không gian màu RGB	12
2.3.2. Không gian màu dựa trên cảm quan.....	13
2.3.3. Không gian màu CIE.....	16
2.4. Các vấn đề cơ bản trong xử lý ảnh	18
2.4.1. Lịch sử về xử lý ảnh	18
2.4.2. Các cấp độ trong xử lý ảnh.....	19
2.4.3. Các ứng dụng của xử lý ảnh.....	20

2.5. Một số phương pháp mã hóa ảnh	23
2.5.1. Mã hóa ảnh dựa trên biến đổi Fourier rời rạc	23
2.5.2. Mã hóa ảnh dựa trên biến đổi cosin rời rạc (DCT)	24
2.5.3. Mã hóa ảnh dựa trên biến đổi sóng rời rạc (DWT)	27

CHƯƠNG 3. NGHIÊN CỨU THUẬT TOÁN MÃ HÓA VÀ GIẢI MÃ BẢO VỆ HÌNH ẢNH Y TẾ..... 29

3.1. Giới thiệu về hình ảnh y tế.....	29
3.2. Thuật toán mã hóa bảo vệ hình ảnh y tế	30
3.2.1. Giới thiệu về các mẫu zigzag, xoay vòng, và hoán vị ngẫu nhiên	30
3.2.2. Ánh xạ logistic hỗn loạn	30
3.2.3. Tách hình ảnh gốc	31
3.2.4. Tạo khóa.....	32
3.2.5. Khuếch tán	33
3.3. Thuật toán giải mã hình ảnh y tế.....	35
3.4. Kết quả mô phỏng trong [1]	35

CHƯƠNG 4. THỰC NGHIỆM THUẬT TOÁN MÃ HÓA BẢO VỆ HÌNH ẢNH Y TẾ VÀ GIẢI MÃ 39

4.1. Mô tả thực nghiệm	39
4.1.1. Kịch bản thực nghiệm.....	39
4.1.2. Giới thiệu về công cụ MATLAB	40
4.2. Tiến hành thực nghiệm	40
4.2.1. Chương trình mã hóa ảnh.....	41
4.2.2. Chương trình giải mã ảnh	44
4.3. Đánh giá kết quả thực nghiệm	46
4.3.1. Phân tích histogram hình ảnh.....	46
4.3.2. Phân tích PSNR.....	47
4.3.3. Entropy.....	48

4.3.4. Phân tích không gian khóa.....	49
4.3.5. Phân tích Different attack	49
PHỤ LỤC	53
Phụ lục 1: Chương trình mã hoá	53
Phụ lục 2: Chương trình giải mã	56
Phụ lục 3: Đánh giá.....	60

DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

Thứ tự	Ký hiệu và từ viết tắt	Nghĩa của ký hiệu và từ viết tắt	Nghĩa tiếng Việt của ký hiệu và từ viết tắt
A	GIF	Graphics Interchange Format	Định dạng trao đổi đồ họa
B	PNG	Portable Network Graphics	Đồ họa mạng di động
C	RGB	Red – Green – Blue	Đỏ- Xanh lá- Xanh dương
D	HEX	Hexadecimal	hệ thập phân
E	bit	Binary Digit	Hệ nhị phân
F	XOR	Exclusive OR	Độc quyền Hoặc
G	MSE	Mean Squared Error	Lỗi bình phương trung bình
H	PSNR	Peak Signal-to-Noise Ratio	Tỷ lệ tín hiệu trên nhiễu cao nhất

DANH MỤC HÌNH ẢNH

Hình 2.1. Ví dụ về ảnh số	3
Hình 2.2. (a) Ảnh gốc, (b) Histogram của hình ảnh	5
Hình 2.3. So sánh hình ảnh vector và raster khi mô tả thế giới thực.....	7
Hình 2.4. Mô tả hình ảnh raster	8
Hình 2.6. Hình ảnh đen trắng chụp MRI não	10
Hình 2.7. Phổ sáng nhìn thấy (độ dài sóng được tính bằng mét)	11
Hình 2.8. Không gian màu RGB.....	12
Hình 2.9. Không gian màu HSL	15
Hình 2.10 Không gian màu HSV	16
Hình 2.11. Không gian màu CIELab	17
Hình 2.12. Làm sắc nét và phục hồi hình ảnh (a) ảnh gốc, (b) hình ảnh phóng to, (c) làm mờ hình ảnh, (d) làm sắc nét hình ảnh, (e) lấy các cạnh của hình ảnh	21
Hình 2.13. Phân tích dữ liệu ảnh từ vệ tinh và máy bay không người lái	22
Hình 2.14. (a) Ảnh gốc (b) Ảnh biên độ (c) Phổ pha	24
Hình 2.15. Ví dụ bảng các hệ số DCT.....	26
Hình 2.16. Phân chia 3 miền tần số thấp, giữa, cao của phép biến đổi DCT	27
Hình 2.17. Biến đổi Wavelet và cấu trúc dải thông.....	28
Hình 3.1. Các vòng tiếp theo của thuật toán trộn Fisher – Yates	31
Hình 3.2. Mẫu zigzag.....	32
Hình 3.3. Sơ đồ phân nhánh của logistic map	33
Hình 3.4. Sơ đồ luồng thuật toán	34
Hình 3.5. Giá trị hệ số tương quan.....	37
Hình 3.6. So sánh giá trị hệ số tương quan giữa thuật toán đề xuất và các thuật toán khác.	37
Hình 3.7. Độ nhảy của thuật toán của chúng tôi đối với khóa. (a) Hình ảnh gốc. (b) Hình ảnh đã mã hóa của (a) bằng cách sử dụng khóa thứ nhất. (c) Hình ảnh đã giải mã của (b) bằng cách sử dụng khóa thứ hai. (d) Hình ảnh đã giải mã của (b) bằng cách sử dụng khóa thứ nhất.....	38

Hình 3.8. Phân tích PSNR	38
Hình 4.1. Hình ảnh thử nghiệm	39
Hình 4.2. Giao diện chương trình	41
Hình 4.3. Nút lựa chọn hình ảnh đầu vào	41
Hình 4.4. Cửa sổ chọn hình ảnh.....	42
Hình 4.5. Hình ảnh sau khi được chọn	43
Hình 4.6. Chọn chức năng giải mã hình ảnh	43
Hình 4.7. Hình ảnh sau khi mã hóa.....	44
Hình 4.8. Hình ảnh được hiển thị sau khi mã hóa	45
Hình 4.9. Hình ảnh hiển thị sau khi giải mã	45
Hình 4.10. Chức năng lưu hình ảnh.....	46
Hình 4.11. Hiển thị sau khi lưu thành công.....	46
Hình 4.12. Phân tích biểu đồ tần suất của hình ảnh xám.....	47

DANH MỤC BẢNG BIỂU

Bảng 4.1. Phân tích PSNR	48
Bảng 4.2. Entropy của hình ảnh đã được mã hóa	49
Bảng 4.3. Giá trị entropy của phương pháp được đề xuất với các phương pháp khác	49
Bảng 4.4. Hiệu suất NPCR	50
Bảng 4.5. So sánh NPCR	50

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Tính cấp thiết của đề tài

Vào năm 2020, theo Cơ quan nghiên cứu Ung thư quốc tế, thuộc tổ chức y tế thế giới WHO đã thống kê trên thế giới có khoảng 19,3 triệu ca ung thư mới và gần 10 triệu ca tử vong do ung thư. Những số liệu thống kê như thế này nhấn mạnh tầm quan trọng của việc tiếp cận các dịch vụ chẩn đoán bằng hình ảnh y tế, để chẩn đoán bệnh sớm khi có khả năng điều trị tốt nhất và giảm thiểu các thủ tục can thiệp phẫu thuật.

Ngoài ra, sử dụng hình ảnh y tế còn giúp bệnh nhân tiết kiệm chi phí chăm sóc sức khỏe. Một nghiên cứu tại trường Y Havard đã phát hiện ra rằng cứ 385 đô la chi cho chụp ảnh y tế sẽ giảm thời gian nằm viện của bệnh nhân xuống 1 ngày, tiết kiệm khoảng 3000 đô la cho mỗi bệnh nhân.

Với sự phát triển của các dịch vụ y tế từ xa như phẫu thuật từ xa, tư vấn từ xa và nhu cầu trao đổi hình ảnh y tế, những dữ liệu y tế này cần được truyền qua một phương tiện truyền thông an toàn để bảo vệ thông tin nhạy cảm của bệnh nhân.

- Nếu hình ảnh y tế truyền tải bị kẻ tấn công lấy được và làm giả mạo sẽ dẫn đến vấn đề chẩn đoán sai.
- Tính bảo mật và toàn vẹn đã trở thành một thách thức quan trọng trong việc truyền tải hình ảnh y tế.
- Các kỹ thuật tiêu chuẩn được sử dụng trong bảo vệ hình ảnh y tế bao gồm: mã hóa, ẩn mã và thủy ấn.

1.2. Lý do chọn đề tài

Trong thời đại số hóa ngày nay, việc bảo vệ thông tin y tế trở nên cực kỳ quan trọng. Hình ảnh y tế chứa thông tin nhạy cảm và quan trọng về sức khỏe của bệnh nhân. Việc mã hóa hình ảnh y tế giúp bảo vệ thông tin này khỏi những người không được phép truy cập. Thuật toán mã hóa hình ảnh y tế là một lĩnh vực nghiên cứu hấp dẫn và thách thức. Việc nghiên cứu và phát triển các thuật toán mới có thể đóng góp cho sự phát triển của lĩnh vực công nghệ số hiện nay. Đề tài này cung cấp cơ hội để bạn tìm hiểu sâu hơn về các thuật toán mã hóa, cũng như phát triển kỹ năng lập trình và giải quyết vấn đề. Kết quả của đề tài này có thể được áp dụng trong thực tế, giúp cải thiện bảo mật thông tin y tế trong các hệ thống y tế số. Đề tài này cũng có thể được chọn để đáp ứng

các yêu cầu của chương trình học, như một phần của một dự án cuối kỳ, luận văn hoặc đồ án tốt nghiệp.

1.3. Mục đích nghiên cứu

- Tìm hiểu về ảnh và các phương pháp mã hóa ảnh
- Tìm hiểu về các mẫu zigzag, xoay vòng, hoán vị ngẫu nhiên, ánh xạ logistic hỗn loạn
- Tìm hiểu về thuật toán mã hóa bảo vệ hình ảnh y tế (ảnh màu và ảnh xám)
- Xây dựng thuật toán và đánh giá

1.4. Phạm vi nghiên cứu

Triển khai thực nghiệm thuật toán mã hóa bảo vệ hình ảnh y tế.

Phương pháp nghiên cứu

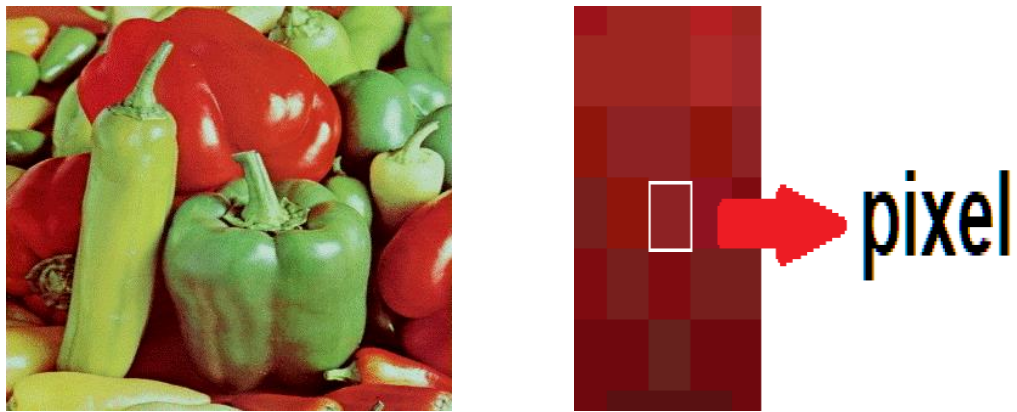
- Phân tích tài liệu: Đọc và phân tích các tài liệu, báo cáo, nghiên cứu liên quan đến thuật toán mã hoá, chuẩn DICOM, và quản lý hình ảnh y tế. Điều này giúp bạn hiểu rõ hơn về lý thuyết và ứng dụng thực tế.
- Phân tích dữ liệu: Sử dụng các công cụ phân tích dữ liệu để khám phá và hiểu rõ hơn về cấu trúc và thông tin trong ảnh DICOM.
- Thực nghiệm: Thực hiện thực nghiệm với các thuật toán mã hoá trên dữ liệu thực tế. Điều này giúp bạn kiểm tra hiệu quả và độ chính xác của thuật toán.
- So sánh và đánh giá: So sánh kết quả của các thuật toán khác nhau, đánh giá ưu và nhược điểm của từng thuật toán để lựa chọn phương pháp tối ưu.
- Xây dựng mô hình: Dựa trên kết quả nghiên cứu, xây dựng mô hình hoạt động cho hệ thống mã hoá thông tin trong ảnh DICOM.
- Kiểm thử: Thực hiện các bài kiểm thử để đảm bảo hệ thống hoạt động đúng và hiệu quả.
- Đánh giá và cải tiến: Dựa trên kết quả kiểm thử, đánh giá hiệu suất của hệ thống và tiến hành cải tiến nếu cần.

CHƯƠNG 2. TỔNG QUAN VỀ ẢNH SỐ VÀ MỘT SỐ PHƯƠNG PHÁP MÃ HÓA ẢNH

2.1. Một số khái niệm về ảnh số

2.1.1. Khái niệm về ảnh số

Ảnh là thông tin về vật thể hay quang cảnh được chiếu sáng mà con người quan sát và cảm nhận được bằng mắt và hệ thống thần kinh thị giác. Một hình ảnh không khác gì hơn một tín hiệu hai chiều. Một hình ảnh có thể được xác định bởi hàm toán học $f(x, y)$ trong đó x và y là hai tọa độ trong không gian hai chiều. Hàm $f(x, y)$ đại diện cho độ sáng trong trường hợp ảnh đơn sắc, ví dụ như ảnh đen trắng hoặc đại diện cho màu sắc trong trường hợp ảnh màu. Hàm này cung cấp giá trị tại mọi điểm ảnh, cho biết độ sáng hoặc màu sắc tại điểm đó.



Hình 2.1. Ví dụ về ảnh số (Nguồn: Tài liệu[a])

Ảnh số là biểu diễn số học của thông tin hình ảnh thu được từ ảnh liên tục thông qua các phép lấy mẫu và lượng tử hóa. Ảnh số thường được biểu diễn dưới dạng một ma trận các điểm ảnh, trong đó mỗi thông tin điểm ảnh thể hiện thông tin về độ sáng hoặc màu sắc tại một vị trí cụ thể trong hình ảnh.

Cách biểu diễn này thường thay đổi tùy theo loại hình ảnh:

Ảnh nhị phân sử dụng 1 bit cho mỗi điểm ảnh, cho biết điểm ảnh đó có xuất hiện trong hình ảnh hay không.

Ảnh đa mức xám sử dụng 8 bit cho mỗi điểm ảnh, cho phép biểu diễn nhiều mức độ sáng khác nhau.

Ảnh màu thường sử dụng 16 hoặc 24 bit cho mỗi điểm ảnh để biểu diễn thông tin về màu sắc.

Hình ảnh được tổ chức dưới dạng các điểm ảnh được gọi là bitmap. Ngoài ra, ảnh số còn có cách biểu diễn khác dưới dạng vector. Biểu diễn dưới dạng vector không sử dụng ma trận các điểm ảnh mà thường bao gồm các thành phần cơ bản như hình tròn, đường thẳng, hoặc đối tượng phức tạp hơn.

Ảnh bitmap thường được sử dụng để biểu diễn các hình ảnh phức tạp, chẳng hạn như hình ảnh chụp từ máy ảnh hoặc hình ảnh có nhiều chi tiết. Khi xử lý ảnh bitmap, cần phải tính toán từng điểm ảnh riêng lẻ, điều này có thể tốn nhiều thời gian. Hình ảnh bitmap thường gặp hạn chế khi thay đổi kích thước (zoom) hoặc áp dụng các phép biến hình, có thể dẫn đến mất thông tin hoặc làm mờ hình ảnh. Định dạng file phổ biến cho ảnh bitmap bao gồm BMP, JPG, PNG và nhiều định dạng hình ảnh khác.

Ảnh vector thường được sử dụng để biểu diễn các hình ảnh đơn giản hoặc các hình dạng đối tượng cơ bản như hình tròn hoặc đường thẳng. Một trong những lợi ích lớn nhất của ảnh vector là tính toán nhanh chóng. Điều này cho phép hình ảnh vector thích hợp cho các ứng dụng yêu cầu sự linh hoạt và hiệu suất cao. Các định dạng file phổ biến cho ảnh vector bao gồm EPS, AI, CDR và DWG.

Hình 2.1 là một ví dụ về hình ảnh kỹ thuật số, trên thực tế hình ảnh này giống như một mảng hai chiều gồm các số nằm trong khoảng từ 0 đến 255. Mỗi số đại diện cho giá trị của hàm $f(x, y)$ tại bất kỳ điểm nào. Mỗi giá trị đại diện cho một pixel riêng lẻ. Kích thước của hình ảnh chính là kích thước của mảng hai chiều này.

Trong thế giới vật chất, bất kỳ đại lượng nào có thể được đo theo thời gian trong không gian hoặc chiều chiều nào khác đều có thể được coi là tín hiệu. Tín hiệu là một hàm toán học và nó truyền tải một số thông tin.

Tín hiệu có thể là tín hiệu một chiều hoặc hai chiều hoặc nhiều chiều hơn. Tín hiệu một chiều là tín hiệu được đo theo thời gian, ví dụ phổ biến là tín hiệu thoại.

Các tín hiệu hai chiều là những tín hiệu được đo trên một số đại lượng vật lý khác. Ví dụ về một tín hiệu hai chiều là một hình ảnh kỹ thuật số.

2.1.2. Khái niệm về pixel và số bit trên pixel

Pixel là phần tử nhỏ nhất của hình ảnh. Với ảnh xám 8-bit (gray scale image), giá trị của pixel nằm trong khoảng từ 0 đến 255. Giá trị của pixel tại bất kỳ thời điểm nào tương ứng với cường độ của các photon ánh sáng chiếu vào điểm đó. Mỗi pixel lưu trữ một giá trị tỷ lệ thuận với cường độ ánh sáng tại vị trí cụ thể đó.

Số bit trên mỗi pixel, thường được gọi là bpp hoặc bits per pixel, nó xác định số lượng bit được sử dụng để biểu diễn mỗi pixel. Số lượng màu khác nhau trong một hình ảnh phụ thuộc vào độ sâu của màu hoặc bit trên mỗi pixel. Ví dụ, một ảnh 8-bit sẽ có 256 mức xám khác nhau, trong khi một ảnh 24-bit có khả năng biểu diễn hàng triệu màu khác nhau.

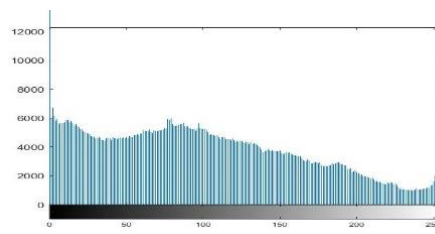
Mật độ pixel trên một hình ảnh chính là độ phân giải của ảnh, là yếu tố quyết định độ chi tiết và độ sắc nét của một hình ảnh. Độ phân giải xác định số lượng pixel trên mỗi đơn vị chiều ngang và chiều dọc của hình ảnh. Độ phân giải càng cao, hình ảnh càng sắc nét và có độ chi tiết cao hơn, trong khi độ phân giải thấp có thể làm cho ảnh mờ và thiếu chi tiết.

2.1.3. Khái niệm về histogram

Histogram là một công cụ trong xử lý hình ảnh, cho phép chúng ta hiểu và phân tích nội dung của một bức ảnh dựa trên phân phối cường độ màu sắc. Nó là một biểu đồ tần số thể hiện sự xuất hiện của các giá trị cường độ màu sắc khác nhau trong hình ảnh.



(a)



(b)

Hình 2.2. (a) Ảnh gốc, (b) Histogram của hình ảnh (Nguồn: Tài liệu[a])

Trong một histogram của hình ảnh, trục x thường biểu thị phạm vi giá trị màu sắc, trục y thể hiện tần suất xuất hiện của các giá trị cường độ này. Có thể dự đoán

về một hình ảnh chỉ bằng cách nhìn vào biểu đồ. Nó cung cấp nhiều thông tin hữu ích chẳng hạn như nó thể hiện được độ tương phản của hình ảnh, tức là khoảng cách giữa các giá trị cường độ. Một hình ảnh có histogram rộng có độ tương phản cao hơn với những hình ảnh có histogram hẹp. Histogram cũng được sử dụng để điều chỉnh độ sáng và tương phản của hình ảnh. Bằng cách phân tích histogram, chúng ta có thể áp dụng các biện pháp điều chỉnh để làm cho ảnh trở nên sắc nét hơn hoặc cân bằng màu sắc. Ngoài ra, histogram đóng vai trò quan trọng trong việc xác định ngưỡng để phân đoạn ảnh thành các vùng khác nhau dựa trên cường độ xám. Nó chủ yếu được sử dụng trong thị giác máy tính như phát hiện biên, định dạng đối tượng, hay nhận dạng vật thể.

2.1.4. Mức xám

Mức xám đại diện cho cường độ hay độ sáng của một điểm ảnh tại một vị trí cụ thể trong hình ảnh. Mức xám có thể được hiểu đơn giản là mức độ đen trắng hoặc cường độ ánh sáng tại một điểm ảnh.

Trong ảnh xám 8-bit, mức xám được biểu thị bằng giá trị số từ 0 đến 255. Giá trị này thể hiện mức độ sáng tại điểm đó, với 0 đại diện cho màu đen tuyệt đối và 255 là màu trắng tuyệt đối. Giữa hai giá trị này có các mức xám khác nhau tạo nên hình ảnh với các mức xám tương phản khác nhau.

Mức xám là một đặc trưng để phân biệt các đối tượng và chi tiết trong hình ảnh. Nó thường được sử dụng trong việc xử lý ảnh để phát hiện biên, trích chọn đặc trưng và thực hiện các phép biến đổi như làm mờ hoặc tăng cường độ tương phản. Hay nói cách khác, mức xám như một thang đo độ sáng, mỗi giá trị biểu thị một mức độ khác nhau của ánh sáng tại một vị trí trên ảnh.

Mức xám có ứng dụng rộng rãi trong nhiều lĩnh vực như xử lý ảnh y tế, nhận dạng biểu đồ, và phân tích hình ảnh tự động.

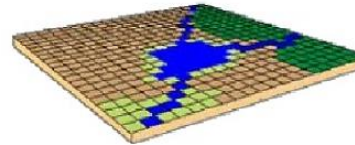
2.2. Phân loại ảnh số

Ảnh số thực chất là biểu diễn số học của hình ảnh trong máy tính, thường là biểu diễn nhị phân, trong đó các điểm ảnh được biểu diễn bằng các giá trị 0 và 1.

Biểu diễn ảnh số bắt nguồn từ việc tách hình ảnh thành các điểm ảnh cơ bản và gán cho mỗi điểm ảnh một giá trị số. Các giá trị này thể hiện độ sáng, màu sắc hoặc các thuộc tính khác của điểm ảnh. Biểu diễn hình ảnh theo cách này giúp máy tính có thể hiểu và xử lý dễ dàng.

Có thể phân ảnh số thành hai loại: ảnh Raster và ảnh Vector.

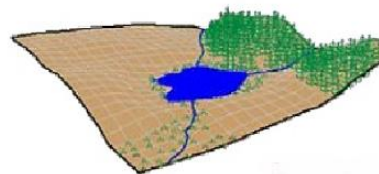
- RASTER



- VECTOR



- Thế giới thực



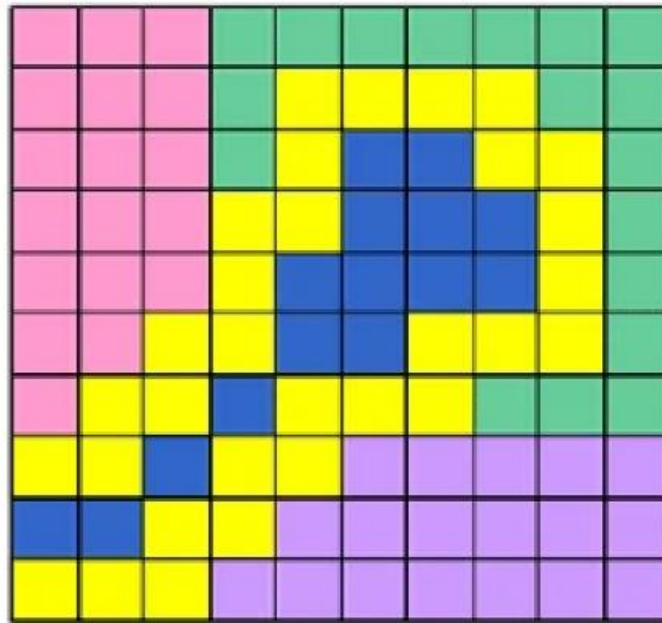
Hình 2.3. So sánh hình ảnh vector và raster khi mô tả thế giới thực(Nguồn: Tài liệu[b])

2.2.1. Ảnh Raster

Ảnh Raster là một tập hợp hữu hạn các giá trị số, gọi là điểm ảnh (pixel). Thông thường, một hình ảnh được chia thành các hàng và cột chứa điểm ảnh. Điểm ảnh là thành phần bé nhất biểu diễn ảnh, có các giá trị số biểu diễn màu sắc, độ sáng, ... của một thành phần trong bức ảnh.

Ảnh Raster thường được thu từ camera, các máy chiếu, chụp, quét,... và chính là đối tượng chính của xử lý ảnh và thị giác máy tính.

Trong y học, Các hình ảnh chụp X-quang, MRI và CT scan thường được biểu diễn dưới dạng ảnh raster để phân tích và chẩn đoán.



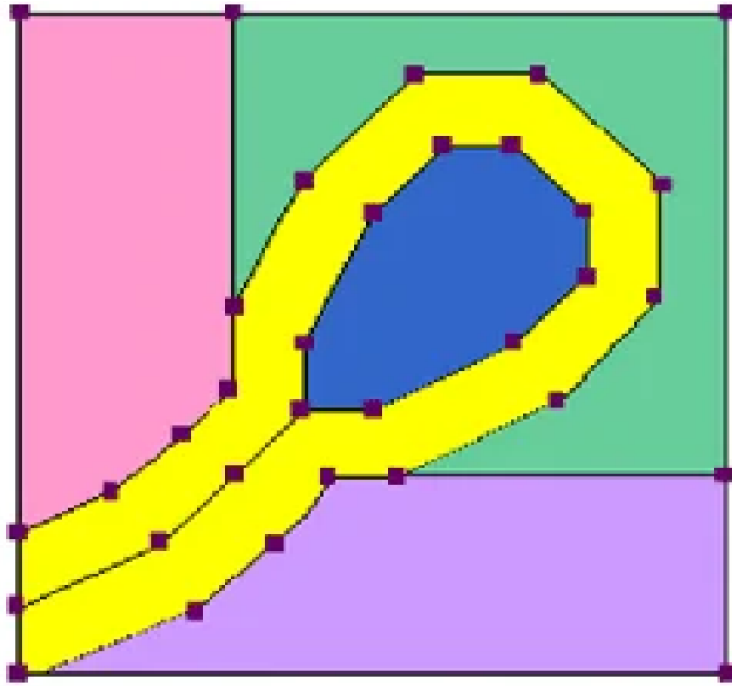
Hình 2.4. Mô tả hình ảnh raster(Nguồn: Tài liệu[b])

2.2.2. Ảnh Vector

Ảnh vector là loại ảnh tạo từ các thành phần đơn giản của hình ảnh như điểm, đường thẳng, hình khối, ... Thay vì được lưu lại thành các ma trận điểm ảnh như ảnh raster, ảnh vector được biểu diễn dưới dạng tọa độ của các thành phần trong ảnh.

Điều này đã tạo nên sự đặc biệt của ảnh vector, khiến nó có thể được kéo dãn, thu nhỏ tùy ý mà không bị vỡ, không xuất hiện răng cưa như ảnh raster. Dữ liệu trong ảnh vector nhỏ, do vậy thường tiết kiệm dung lượng hơn ảnh raster. Tuy thế, màu sắc trong ảnh vector nhìn không thật, sắc độ ít tinh tế hơn ảnh raster.

Thông thường, người ta sử dụng ảnh vector trong thiết kế các logo, banner, giao diện đồ họa, ... Loại ảnh này gần như không xuất hiện khi đề cập đến xử lý ảnh hay thị giác máy tính.

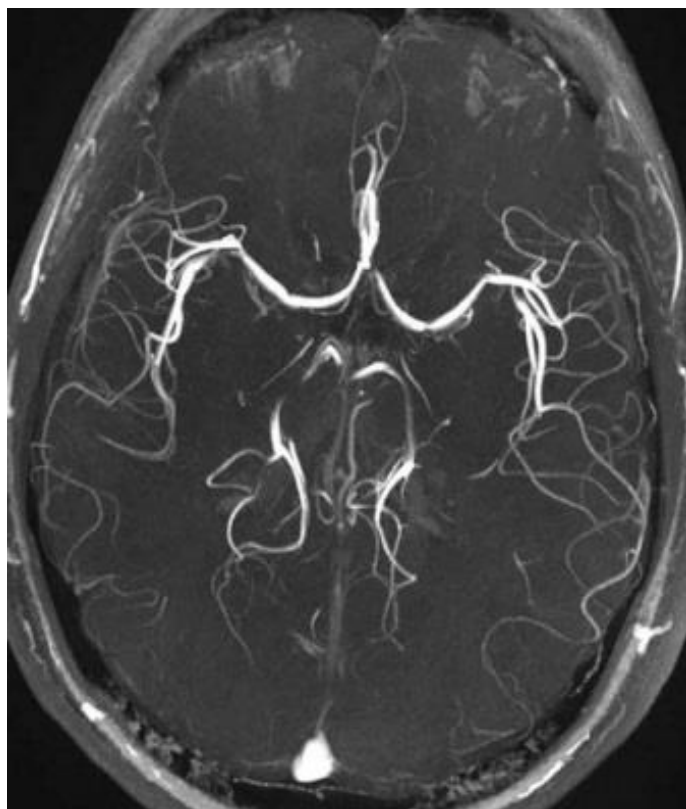


Hình 2.5. Mô tả hình ảnh vector (Nguồn: Tài liệu[b])

Tùy vào giá trị dùng để biểu diễn điểm ảnh, có thể chia ảnh số thành hai loại: ảnh đen trắng và ảnh màu.

2.2.3. Ảnh đen trắng

Ảnh đen trắng là loại ảnh chỉ bao gồm hai màu đen và trắng, không chứa các màu khác. Thực tế ảnh đen trắng lại được phân thành hai loại khác nhau là ảnh nhị phân và ảnh đa cấp xám. Người ta phân mức ảnh đen trắng của ảnh thành L mức. Nếu L bằng 2 nghĩa là chỉ có hai mức 0 và 1, hay còn gọi được là ảnh nhị phân. Nếu L lớn hơn 2 là ảnh đa cấp xám.



Hình 2.6. Hình ảnh đen trắng chụp MRI não(Nguồn: Tài liệu[b])

Nói cách khác, mỗi điểm ảnh nhị phân được mã hóa dựa trên 1 bit còn ảnh xám 256 mức được mã hóa trên 8 bit (1 byte). Với ảnh đen trắng, nếu dùng 8 bit để biểu diễn mức xám thì có thể biểu diễn được 256 mức xám. Mỗi mức xám sẽ được biểu diễn dưới dạng một số nguyên nằm trong khoảng từ 0 đến 255. Trong đó mức 0 biểu diễn cho mức cường độ tối nhất, và 255 biểu diễn cho mức cường độ sáng nhất.

Ảnh nhị phân khá đơn giản, các phần tử ảnh có thể được coi như các phần tử logic. Nó thường được ứng dụng theo logic nhằm phân biệt đối tượng ảnh với nền hoặc phân biệt điểm biên với điểm khác.

Ảnh đen trắng thường được sử dụng trong các tác vụ y tế, xử lý hình ảnh và phân tích dữ liệu.

2.2.4. Ảnh màu

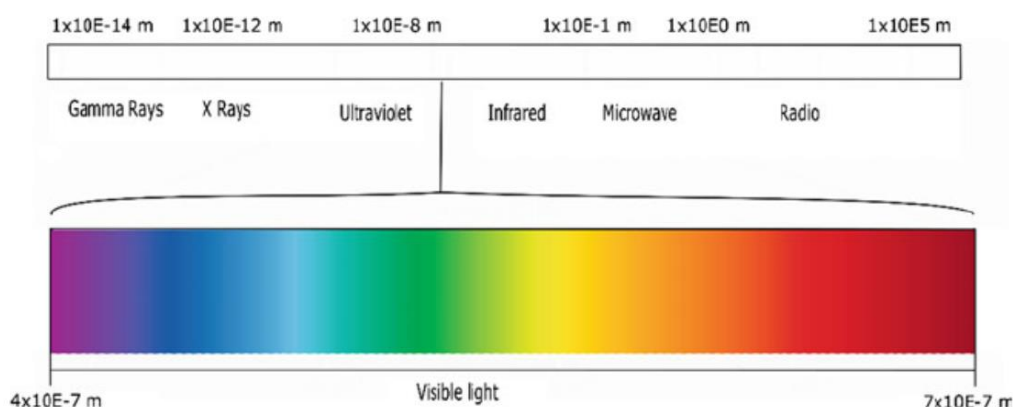
Tùy theo từng trường hợp mà ảnh màu được biểu diễn bằng những không gian màu khác nhau. Không gian màu chính là mô hình toán học dùng để mô tả các màu sắc trong thực tế dưới dạng số học. Không có mô hình nào có thể biểu diễn đầy đủ

khía cạnh của màu. Do đó phải sử dụng những mô hình khác nhau để mô tả những tính chất được nhận biết khác nhau của màu.

Phổ biến nhất có thể kể để không gian màu RGB được ứng dụng để sản xuất màn hình ti vi. Hoặc không gian màu HSV mô tả màu dựa trên nhận thức của con người.

2.3. Các không gian màu biểu diễn ảnh số

Màu sắc là một trong những yếu tố quan trọng trong thị giác, vì nó thường được sử dụng để phân biệt và nhận dạng thông tin. Từ góc độ vật lý, màu sắc là kết quả của sự phản xạ của một phần ánh sáng trên một đối tượng. Về cơ bản nó chỉ là một sóng điện từ trong một phạm vi tần số nhất định. Bước sóng dài nhất tạo ra màu đỏ và bước sóng ngắn nhất tạo ra màu tím. Các màu khác tồn tại ngoài phạm vi phổ điện từ được gọi là màu không thể nhìn thấy (bức xạ hồng ngoại và tia cực tím).



Hình 2.7. Phổ sáng nhìn thấy (độ dài sóng được tính bằng mét) (Nguồn: Tài liệu[c])

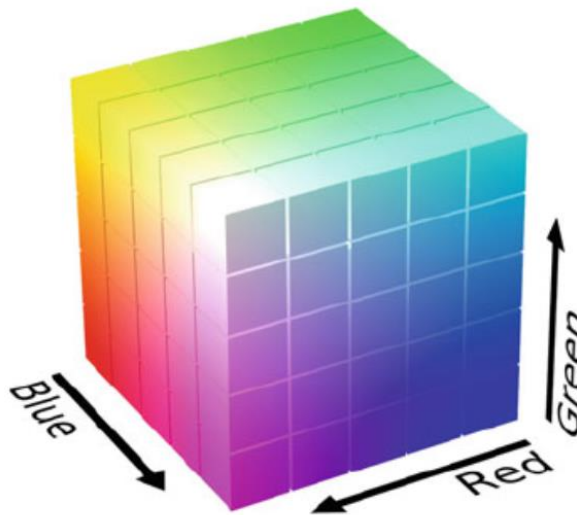
Vì việc biểu diễn màu sắc dưới dạng một tần số sóng cụ thể với một biên độ và chiều dài cụ thể không phải lúc nào cũng dễ dàng hoặc tự nhiên đối với con người, nên các hệ thống giúp mô tả màu sắc theo cách tự nhiên và có trật tự hơn đã được tạo ra. Những hệ thống mô tả này được gọi là các mô hình màu.

Một mô hình màu thực chất là một tập hợp các phương trình hoặc các quy tắc được sử dụng để tính toán một màu cụ thể và có khả năng mô tả một loạt sắc thái từ phổ điện từ màu sắc. Không gian màu được định nghĩa như là tất cả các tông màu có thể được tạo ra bằng một mô hình màu.

2.3.1. Không gian màu RGB

Không gian màu tổ hợp là tập hợp các màu được tạo ra bằng cách kết hợp các màu chính. Các màu phổ biến nhất là đỏ, xanh dương và xanh lá cây. Ban đầu, các màu này được tạo ra bằng cách kết hợp hai hoặc nhiều tia sáng thành một bề mặt tối. Không gian màu được tạo ra bằng cách sử dụng kỹ thuật này gọi là RGB. Một biến thể của không gian màu tổ hợp là không gian màu trừ, thay vì thêm các tia sáng, nó chặn hoặc trừ đi một số tần số từ một tia sáng chứa toàn bộ phạm vi tần số từ phổ sáng nhìn thấy.

Mô hình RGB được sử dụng rộng rãi nhất trong đồ họa máy tính. Nhiều công trình sử dụng mô hình này như một cơ sở cho việc lọc màu, phân đoạn và phân tích trong xử lý hình ảnh. Sự dễ sử dụng và mô hình trực quan để tạo và điều chỉnh màu sắc khiến nó trở thành sự lựa chọn lý tưởng cho việc vẽ và tô màu.



Hình 2.8. Không gian màu RGB(Nguồn: Tài liệu[c])

Tuy nhiên, cũng có một số nhược điểm khi được sử dụng trong xử lý hình ảnh. Mô hình này tạo ra một không gian phi tuyến tính và không liên tục, là cho việc theo dõi sự thay đổi trong màu sắc khó khăn do sự không liên tục. Vấn đề khác của mô hình RGB là màu sắc dễ bị ảnh hưởng bởi thay đổi của ánh sáng, làm cho việc theo dõi và phân tích màu sắc trở nên khó khăn.

Mặc dù tồn tại các vấn đề trên, không gian màu được tạo ra từ nó vẫn được ưa chuộng trong hầu hết các nhiệm vụ tính toán. Một trong những lý do đó là hầu

hết phần cứng và phần mềm máy tính được phát triển dưới mô hình này, vì vậy nó có thể được sử dụng trực tiếp mà không cần công việc bổ sung.

Normalized RGB Mô hình này là một biến thể từ RGB, và cơ bản được tạo ra với tiền đề bảo vệ mô hình màu khỏi sự thay đổi về ánh sáng. Ý tưởng chính đằng sau biến thể này là màu sắc được hình thành bằng cách sử dụng một tỷ lệ nhất định của ba màu chính từ mô hình, không phải là một lượng cố định cho mỗi màu. Vì vậy, tổng tỷ lệ của tất cả các màu phải bằng 100%. Quá trình chuẩn hóa này rất đơn giản và có thể được tính toán dễ dàng (Công thức 1-3). Cơ bản, mỗi phương trình được sử dụng để biến đổi RGB thành RGB chuẩn hóa đều thu được tỷ lệ của mỗi màu. Một ưu điểm khác của mô hình này là bất kỳ màu nào cũng có thể được mô tả chỉ bằng hai màu thay vì ba. Điều này có thể thực hiện được bởi vì màu thứ ba luôn luôn là sự khác biệt giữa một màu và tổng của hai màu còn lại.

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B}$$

Biến thể này chắc chắn giảm bớt tác động tiêu cực của sự thay đổi ánh sáng như bóng hoặc ánh sáng sáng, nhưng đổi lại nó giảm khả năng phát hiện đối tượng. Điều này là do sự mất màu sắc mà cùng một ánh sáng cung cấp.

2.3.2. Không gian màu dựa trên cảm quan

Không gian màu dựa trên cảm quan được tạo ra từ các mô hình xử lý màu sắc theo một cách gần gũi với con người. Để đạt được mục tiêu này, mọi màu sẽ được biểu diễn bằng một sắc thái hoặc màu sắc cụ thể, một mức độ bão hòa cho màu sắc đó và lượng ánh sáng có sẵn hoặc ánh sáng chiếu vào.

a. Mô hình HSL

Mô hình này có thể tính toán từ mô hình tiêu chuẩn RGB bằng cách sử dụng một bộ phương trình (Công thức 4-6). Lợi thế chính của việc sử dụng mô hình này là sự không thay đổi trước các thay đổi ánh sáng, vì ánh sáng được bao quanh trong

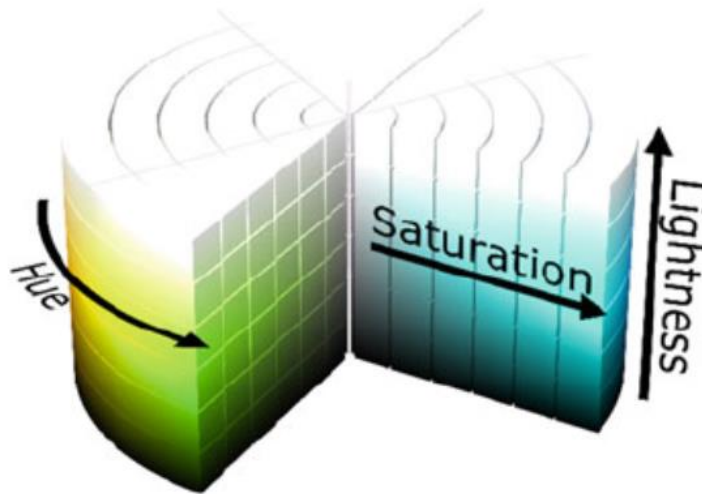
thành phần độ sáng của mô hình. Đặc điểm khác của mô hình này là có thể thấy trong sự thay đổi về màu sắc, chúng liên tục và tuyến tính. Thành phần màu sắc thường được biểu diễn bằng đơn vị góc, từ 0 đến 359°, vì mỗi tông màu có một tông màu trước và tông màu sau theo một mô hình tuần hoàn và có thể được hiển thị trong biểu đồ vòng tròn. Đặc điểm cuối cùng của mô hình HSL là biểu đồ hình học cho phép biểu diễn dễ dàng từng tông màu bằng cách thay đổi độ sáng và độ bão hòa của nó.

$$H = \begin{cases} 60 * \left(\frac{G - B}{\max(R, G, B) - \min(R, G, B)} \right) & R = \max(R, G, B) \\ 60 * \left(\frac{2 + (B - R)}{\max(R, G, B) - \min(R, G, B)} \right) & G = \max(R, G, B) \\ 60 * \left(\frac{4 + (R - G)}{\max(R, G, B) - \min(R, G, B)} \right) & B = \max(R, G, B) \end{cases}$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{1 - |\max(R, G, B) + \min(R, G, B) - 1|}$$

$$L = \frac{\max(R, G, B) + \min(R, G, B)}{2}$$

Mặc dù mô hình HSL có vẻ lý tưởng, nó cũng gặp một số vấn đề; vấn đề đầu tiên xảy ra khi giá trị tối đa và tối thiểu cho RGB giống nhau, tương ứng với các tông màu xám. Trong tình huống này, không có giá trị xác định cho tông màu, mà thường các thư viện phần mềm hiện có sẽ đặt thành giá trị không (màu Đỏ). Điều này dẫn đến một số hiểu lầm trong việc giải mã màu. Các vấn đề khác xảy ra khi tổng giá trị tối đa và tối thiểu là hai; trong tình huống này, độ bão hòa không được xác định.



Hình 2.9. Không gian màu HSL (Nguồn: Tài liệu[c])

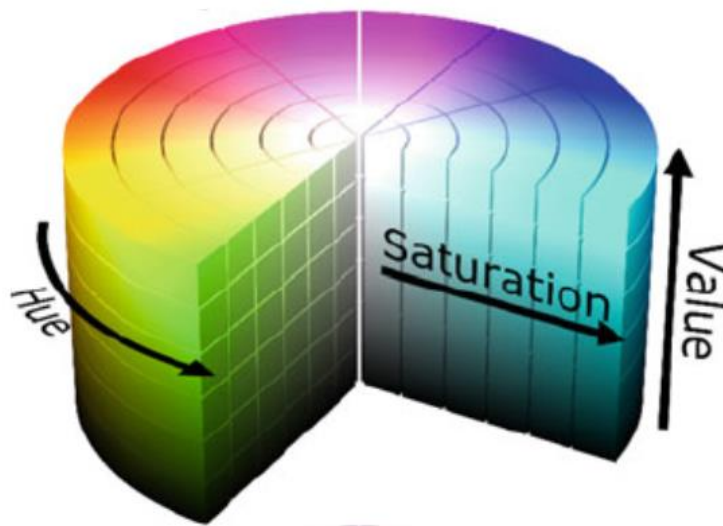
b. Mô hình HSV

Mô hình HSV (Hue Saturation Value) tương tự như HSL nhưng khác nhau trong cách hiểu độ bão hòa màu và ánh sáng. Trong đó Hue là bước sóng của ánh sáng, Saturation là độ bão hòa và Bright (Value) là cường độ hay độ chói ánh sáng.

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)}$$

$$V = \max(R, G, B)$$

Các vấn đề trong mô hình HSL cũng tồn tại trong mô hình HSV, có cùng tình huống không xác định màu sắc và độ bão hòa, vấn đề này nảy sinh khi giá trị tối đa cho RGB là 0 (màu đen).



Hình 2.10 Không gian màu HSV (Nguồn: Tài liệu[c])

Đôi khi HSV được ưa chuộng do biểu diễn thường tự nhiên hơn so với HSL, cho phép sự thay đổi màu sắc tốt hơn.

2.3.3. Không gian màu CIE

CIE (The international Commission on Illumination) là một tổ chức phi lợi nhuận trên toàn cầu, được thành lập với mục đích tổng hợp và chia sẻ thông tin liên quan đến khoa học và nghệ thuật ánh sáng, màu sắc và công nghệ hình ảnh. CIE đã tạo ra các mô hình màu đầu tiên dựa chủ yếu vào khía cạnh vật lý của ánh sáng. Trong số các mô hình đáng chú ý nhất như CIE RGB CIE XYZ và CIELab, các mô hình này mô tả cách màu sắc có thể được hình thành trên một mặt phẳng với một mức độ chiếu sáng cụ thể.

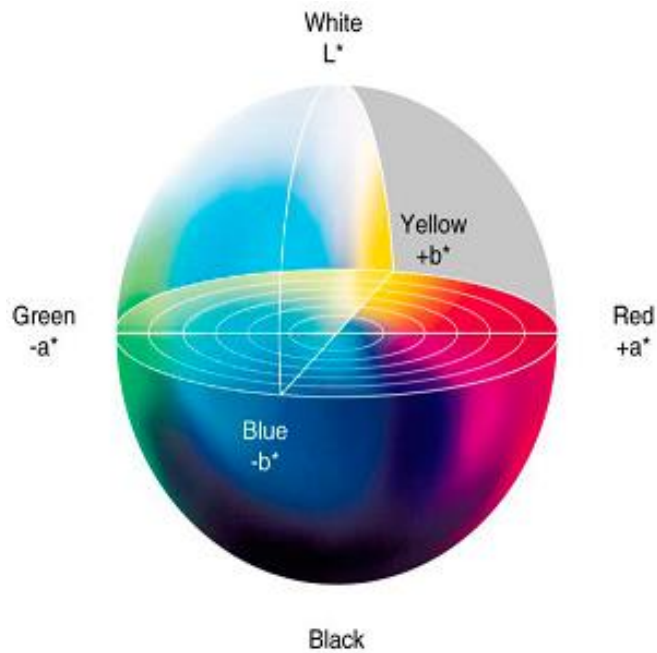
CIE XYZ và CIE RGB được tính toán bằng cách sử dụng bước sóng ánh sáng từ biểu diễn vật lý của màu sắc, trong khi CIELab thu được một cách gián tiếp từ CIE XYZ.

Lab (còn gọi là $L * a * b$) là một trong những mô hình phổ biến nhất trong các mô hình CIE. Mô hình Lab bao gồm một thành phần Lightness (đo độ sáng hoặc độ tối của màu sắc) và hai thành phần màu sắc (a và b). Mô hình màu này có thể biểu thị một phạm vi màu sắc lớn hơn so với RGB và thường được sử dụng để cải thiện hình ảnh màu. Mô hình Lab có thể thu được từ mô hình CIE XYZ bằng cách sử dụng công thức dưới đây, trong đó $X_n = 95.047$, $Y_n = 100.0$ và $Z_n = 108.883$ là giá trị tham chiếu ba trị tiêu chuẩn cho mô hình CIE XYZ.

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right)$$



Hình 2.11. Không gian màu CIELab (Nguồn: Tài liệu[c])

Mô hình CIELab có thể biểu diễn các màu sắc mà các mô hình khác không xử lý được. Theo lý thuyết nó có thể biểu diễn một số lượng vô hạn các kết hợp màu sắc. Tuy nhiên, việc tạo ra nó là quá trình phức tạp, và không gian màu được tạo ra không tự nhiên như trong RGB hoặc các không gian màu theo mô hình dựa trên cảm quan.

2.4. Các vấn đề cơ bản trong xử lý ảnh

2.4.1. Lịch sử về xử lý ảnh

Xử lý ảnh số là một lĩnh vực quan trọng trong khoa học máy tính, phát triển theo thời gian từ các ứng dụng đầu tiên đến những tiến bộ phức tạp và ứng dụng đa dạng mà chúng ta thấy ngày nay.

Lịch sử của xử lý ảnh số có nguồn gốc từ hai ứng dụng chính: nâng cao chất lượng thông tin hình ảnh và xử lý số liệu cho máy tính. Ứng dụng đầu tiên của xử lý ảnh số xuất hiện vào năm 1920, khi thông tin ảnh được truyền qua cáp Bartlane từ London đến New York. Khi đó, việc mã hóa và khôi phục dữ liệu ảnh đã giúp rút ngắn thời gian truyền ảnh từ 1 tuần xuống còn 3 tiếng.

Máy in điện tử McFarlane đã tạo ra ảnh số đầu tiên vào năm 1921 từ băng mã hóa. Tại năm 1922, ảnh số tiếp theo được tạo ra từ card đục lỗ sau khi truyền qua Đại Tây Dương. Mặc dù có một số lỗi nhỏ, nhưng ảnh đã có thể nhìn thấy được. Năm 1929, một cơ hội mới mở ra khi ảnh 15 cấp độ xám được truyền từ Luân Đôn đến New York.

Tuy trong thời kỳ này người ta chỉ nói về ảnh số mà chưa đề cập đến việc xử lý ảnh số, nhưng những cơ sở đầu tiên cho sự phát triển của lĩnh vực này đã được đặt nền móng.

Một bước tiến quan trọng trong lịch sử xử lý ảnh là vào những năm 1960 và 1970, khi ngành khoa học này song song phát triển với việc khám phá không gian và các ứng dụng trong y học, theo dõi tài nguyên trái đất và thiên văn học. Vào năm 1964, ảnh của mặt trăng được chụp bởi tàu vũ trụ Mỹ Ranger 7 và sau đó được xử lý để chỉnh méo.

Xử lý ảnh số đã trải qua một bước tiến dài và phức tạp trong nhiều ngành khoa học khác nhau. Từ việc nâng cao chất lượng hình ảnh, việc xử lý số liệu và theo dõi tài nguyên đến việc nghiên cứu thiên văn học và khoa học không gian, xử lý ảnh số đã tiến xa hơn để đáp ứng các yêu cầu ngày càng tăng về khả năng xử lý và ứng dụng.

Từ việc cải thiện hình ảnh và hiểu biết về vũ trụ đến việc áp dụng trong y học và ngành công nghiệp, lịch sử của xử lý ảnh số thể hiện sự phát triển không ngừng của công nghệ và sự ứng dụng rộng rãi của nó trong cuộc sống hàng ngày và nhiều lĩnh vực chuyên ngành khác nhau.

2.4.2. Các cấp độ trong xử lý ảnh

Xử lý ảnh là quá trình xử lý và biến đổi hình ảnh để trích xuất thông tin, tăng cường chất lượng, và thậm chí hiểu biết về nội dung của ảnh. Quá trình này được chia thành các cấp độ khác nhau, mỗi cấp độ có mục tiêu và nhiệm vụ riêng biệt. Dưới đây là các cấp độ trong xử lý ảnh:

a. Cấp Độ 0: Thu Nhận Ảnh

Cấp độ này liên quan đến việc thu thập dữ liệu hình ảnh từ các nguồn khác nhau. Quá trình này bao gồm việc lấy mẫu, lượng tử hóa, và nén ảnh. Trong giai đoạn này, dữ liệu hình ảnh chưa được xử lý hay biến đổi.

b. Cấp Độ 1: Xử Lý Ảnh Sang Ảnh

Ở cấp độ này, ảnh gốc được biến đổi để tạo ra ảnh mới có chất lượng cao hơn hoặc thông tin hữu ích hơn. Các công việc tại cấp độ này như:

Tăng Cường Ảnh: Quá trình này bao gồm việc tăng cường độ tương phản, làm sáng hoặc làm tối ảnh, cải thiện độ nét và loại bỏ nhiễu.

Khôi Phục Ảnh: Các kỹ thuật khôi phục ảnh tập trung vào việc loại bỏ các biến đổi hoặc lỗi có thể xuất hiện trong quá trình thu thập dữ liệu như méo mó, nhiễu, hoặc hỏng hóc.

Phân Đoạn Ảnh: Quá trình này tập trung vào việc chia ảnh thành các vùng có ý nghĩa khác nhau. Phân đoạn có thể là việc tách vùng đối tượng khỏi nền hoặc chia ảnh thành các vùng với các thuộc tính tương tự.

c. Cấp Độ 2: Xử Lý Ảnh Sang Tham Số

Ở cấp độ này, thông tin từ ảnh được trích xuất và biểu diễn dưới dạng các tham số. Các công việc tại cấp độ này bao gồm:

Trích Chọn Đặc Trưng: Quá trình này tập trung vào việc chọn ra các đặc điểm quan trọng của ảnh để mô tả chúng một cách tốt nhất. Điều này thường liên quan đến việc chọn các thuộc tính như màu sắc, kích thước, hình dạng, v.v.

Lựa Chọn Đặc Trưng: Trong bước này, chọn ra các đặc trưng quan trọng từ tập các đặc trưng đã trích xuất. Điều này giúp giảm thiểu thông tin không cần thiết và tăng hiệu suất tính toán.

d. Cấp Độ 3: Từ Tham Số Đến Quyết Định

Ở cấp độ này, thông tin tham số đã trích xuất được sử dụng để đưa ra quyết định hoặc tạo ra hiểu biết về nội dung của ảnh. Các công việc tại cấp độ này bao gồm:

Nhận Dạng: Quá trình này liên quan đến việc nhận biết và phân loại các đối tượng hoặc mẫu trong ảnh. Điều này có thể là nhận dạng khuôn mặt, đối tượng, vật thể, v.v.

Giải Thích: Tại cấp độ này, thông tin đã trích xuất được sử dụng để tạo ra hiểu biết về nội dung của ảnh. Điều này có thể liên quan đến việc tạo ra mô tả, diễn giải về tình huống, v.v.

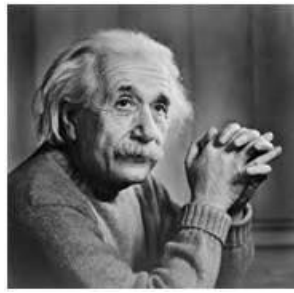
2.4.3. Các ứng dụng của xử lý ảnh

Xử lý ảnh đã có những đóng góp quan trọng và đa dạng trong nhiều lĩnh vực khác nhau, từ khoa học đến công nghiệp và cuộc sống hàng ngày. Dưới đây là một số ứng dụng quan trọng của xử lý ảnh:

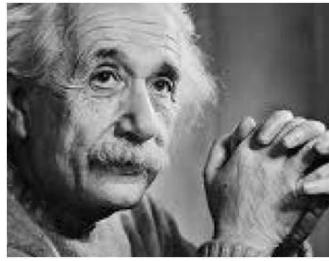
a. Làm sắc nét và phục hồi hình ảnh

Làm sắc nét và phục hồi hình ảnh ở đây đề cập đến việc xử lý các hình ảnh đã được chụp từ máy ảnh hiện đại để làm cho chúng trở thành hình ảnh đẹp hơn hoặc xử lý những hình ảnh đó theo cách để đạt được kết quả mong muốn. Nó đề cập đến những gì Photoshop thường làm.

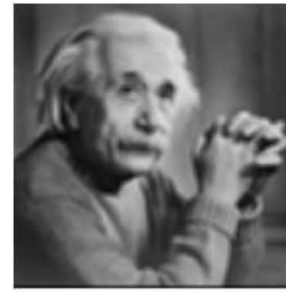
Điều này bao gồm Thu phóng, làm mờ, làm sắc nét, chuyển đổi thang xám sang màu, phát hiện các cạnh và ngược lại, Truy xuất hình ảnh và Nhận dạng hình ảnh. Các ví dụ phổ biến là:



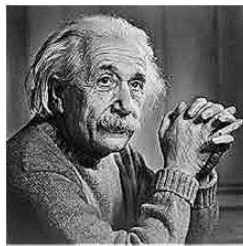
(a)



(b)



(c)



(d)



(e)

Hình 2.12. Làm sắc nét và phục hồi hình ảnh (a) ảnh gốc, (b) hình ảnh phóng to, (c) làm mờ hình ảnh, (d) làm sắc nét hình ảnh, (e) lấy các cạnh của hình ảnh (Nguồn: Tài liệu[d])

b. Lĩnh vực y tế

Y tế là một trong những lĩnh vực quan trọng của xử lý ảnh. Nó được sử dụng trong việc chẩn đoán hình ảnh y khoa như X-quang, MRI, CT-scan để giúp bác sĩ chẩn đoán và điều trị bệnh. Xử lý ảnh cũng được sử dụng để theo dõi sự phát triển của tế bào, phân tích hình ảnh về gen và nghiên cứu về dược phẩm.

c. Hình ảnh vệ tinh

Trong lĩnh vực viễn thám, khu vực trái đất được quét bởi một vệ tinh hoặc từ một mặt đất rất cao và sau đó nó được phân tích để thu được thông tin về nó. Một ứng dụng cụ thể của xử lý ảnh số trong lĩnh vực viễn thám là phát hiện các thiệt hại về cơ sở hạ tầng do động đất gây ra.

Vì phải mất nhiều thời gian hơn để nắm bắt thiệt hại, ngay cả khi tập trung vào những thiệt hại nghiêm trọng. Vì khu vực bị ảnh hưởng bởi trận động đất đôi khi quá rộng nên không thể kiểm tra bằng mắt người để ước tính thiệt hại, đó là thủ tục rất khó khăn và tốn thời gian. Vì vậy, một giải pháp cho điều này là xử lý hình

ảnh kỹ thuật số. Một hình ảnh của khu vực bị ảnh hưởng được chụp từ mặt đất và sau đó nó được phân tích để phát hiện các loại thiệt hại khác nhau do trận động đất gây ra.



*Hình 2.13. Phân tích dữ liệu ảnh từ vệ tinh và máy bay không người lái
(Nguồn: Tài liệu[d])*

d. Truyền và mã hóa

Hình ảnh đầu tiên được truyền qua đây là từ London đến New York thông qua cáp ngầm. Bức ảnh được gửi mất ba giờ để đến nơi này đến nơi khác.

Ngày nay chúng ta có thể xem nguồn cấp dữ liệu video trực tiếp hoặc cảnh quay camera quan sát trực tiếp từ lục địa này sang lục địa khác chỉ với độ trễ vài giây. Điều đó có nghĩa là rất nhiều công việc đã được thực hiện trong lĩnh vực này. Lĩnh vực này không chỉ tập trung vào truyền tải mà còn về mã hóa. Nhiều định dạng khác nhau đã được phát triển cho băng thông cao hoặc thấp để mã hóa ảnh và sau đó truyền trực tiếp qua internet, v.v.

e. Robot và tự động hóa

Trong lĩnh vực robot và tự động hóa, xử lý ảnh đóng vai trò quan trọng trong việc điều khiển và định vị robot. Các hệ thống xử lý ảnh giúp robot nhận biết môi trường, tránh vật cản, và thực hiện nhiều tác vụ phức tạp.

f. Xử lý màu

Xử lý màu bao gồm xử lý các hình ảnh có màu và các không gian màu khác nhau được sử dụng. Ví dụ: mô hình màu RGB, YCbCr, HSV. Nó cũng liên quan đến việc nghiên cứu truyền tải, lưu trữ và mã hóa những hình ảnh màu này.

g. Giám Sát và Phát Hiện Chuyển Động

Ứng dụng của xử lý ảnh trong việc giám sát và phát hiện chuyển động rất rộng rãi. Nó được sử dụng trong việc theo dõi giao thông, giám sát an ninh, giám sát môi trường tự nhiên, nhận biết hành vi người và vật thể trong các hình ảnh hoặc video.

h. Xử lý video

Video chính là chuyển động rất nhanh của hình ảnh. Chất lượng của video phụ thuộc vào số khung hình/hình ảnh mỗi phút và chất lượng của từng khung hình được sử dụng. Xử lý video bao gồm giảm nhiễu, tăng cường chi tiết, phát hiện chuyển động, chuyển đổi tốc độ khung hình, chuyển đổi tỷ lệ khung hình, chuyển đổi không gian màu, v.v.

2.5. Một số phương pháp mã hóa ảnh

2.5.1. Mã hóa ảnh dựa trên biến đổi Fourier rời rạc

a. Biến đổi DFT một chiều

Biến đổi Fourier 1-D cho tín hiệu thời gian rời rạc $f(kT)$ theo công thức:

$$F(n) = \sum_{k=0}^{N-1} f(kT) e^{\frac{-j2\pi}{N}nk}$$

Công thức này có thể viết dưới dạng

$$F(n) = \sum_{k=0}^{N-1} f(k) W_N^{-nk}$$

$f(k) = f(kT)$ và $W_N = e^{\frac{-j2\pi}{N}}$. W_N được gọi là hạt nhân của phép biến đổi.

Tổng quát, $F(n)$ có dạng:

$$F(n) = A(n) e^{j\phi(n)}$$

Kí hiệu $A(n)$, $\phi(n)$ gọi là phổ khuếch đại và phổ pha của $F(n)$

Biến đổi ngược DFT

Hàm $f(k)$ là biến đổi ngược DFT của $F(n)$ cho bởi theo biểu thức:

$$f(k) = \frac{1}{N} \sum_{n=0}^{N-1} F(n) e^{j\frac{2\pi}{N}nk}$$

Chứng minh : Từ định nghĩa của DFT

b. Biến đổi DFT hai chiều

Một DFT hai chiều của tín hiệu lấy mẫu hai chiều $h(k_1, k_2)$ cho bởi:

$$H(n_1, n_2) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} h(k_1, k_2) e^{\frac{-j2\pi}{N(n_1 k_1 + n_2 k_2)}} = DFT\{h(k_1, k_2)\}$$

Ở đây, $n_1 = 0, 1, 2, \dots, N-1$ và $n_2 = 0, 1, 2, \dots, N-1$

Biểu thức $e^{\frac{-j2\pi}{N(n_1 k_1 + n_2 k_2)}}$ trong hai dấu tổng gọi là hạt nhân của phép biến đổi.

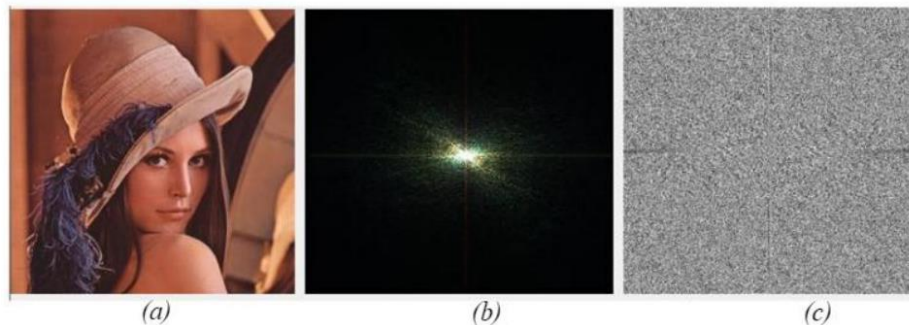
$H(n_1, n_2)$, trong trường hợp tổng quát, đầy đủ có thể biểu diễn theo:

$$H(n_1, n_2) = A(n_1, n_2) e^{j\phi(n_1, n_2)}$$

Trong không gian ba chiều $A(n_1, n_2)$ và $\phi(n_1, n_2)$ nằm tại vị trí n_1 và n_2 gọi là phổ tần và phổ pha của $H(n_1, n_2)$.

Hàm $h(k_1, k_2)$ là biến đổi ngược 2-D DFT (IDFT) của hàm $H(n_1, n_2)$ và được cho bởi biểu thức:

$$h(k_1, k_2) = \frac{1}{N^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} H(n_1, n_2) e^{\frac{j2\pi}{N(n_1 k_1 + n_2 k_2)}}$$



Hình 2.14. (a) Ảnh gốc (b) Ảnh biên độ (c) Phổ pha (Nguồn: Tài liệu[d])

2.5.2. Mã hóa ảnh dựa trên biến đổi cosin rời rạc (DCT)

Biến đổi cosin rời rạc DCT được đưa ra bởi Ahmed và các đồng nghiệp vào năm 1974. Từ đó đến nay, nó được sử dụng phổ biến trong nhiều kỹ thuật xử lý ảnh

số nói riêng và xử lý tín hiệu số nói chung. Nó được sử dụng chuẩn nén JPEG để mã hóa ảnh tĩnh và chuyển MPEG để mã hóa ảnh động.

Biến đổi DCT hai chiều tổng quát là biến đổi trong khối hai chiều bất kỳ $M \times N$. Sau đây trình bày công thức biến đổi DCT 2 chiều trên khối kích thước 8×8 được sử dụng nhiều nhất hoặc 16×16 .

Công thức biến đổi DCT thuận từ $I(k,l) \rightarrow I(u,v)$

$$I(u, v) = \frac{C(u)C(v)}{4} \sum_{k=0}^7 \sum_{l=0}^7 I(k, l) \cos\left(\frac{(2k+1)u\pi}{16}\right) \cos\left(\frac{(2l+1)v\pi}{16}\right)$$

$I(u,v)$ được gọi là hệ số DCT và là số thực.

Công thức biến đổi ngược IDCT từ $I(u,v) \rightarrow I(k,l)$

$$I(k, l) = \sum_{u=0}^7 \sum_{v=0}^7 \frac{C(u)C(v)}{4} I(u, v) \cos\left(\frac{(2k+1)u\pi}{16}\right) \cos\left(\frac{(2l+1)v\pi}{16}\right)$$

Ở đây $0 \leq k, l, u, v \leq 7$

Phép biến đổi DCT ảnh hai chiều thể hiện đặc tính nội dung về tần số của thông tin ảnh. Hầu hết các thuật toán, ảnh gốc được chia thành các khối ma trận ảnh 8×8 . Áp dụng biến đổi DCT cho mỗi khối ta sẽ thu được khối 8×8 chứa các hệ số DCT. Gọi $C_b(j, k)$ là giá trị các hệ số trong đó b là số thứ tự của khối, (j, k) là vị trí của hệ số. Hệ số đầu tiên $C_b(0,0)$ được gọi là Dc và chứa thông tin độ sáng của khối đó. Các hệ số còn lại biểu diễn cho các thành phần tần số cao theo hướng ngang và theo hướng thẳng đứng gọi là hệ số AC.

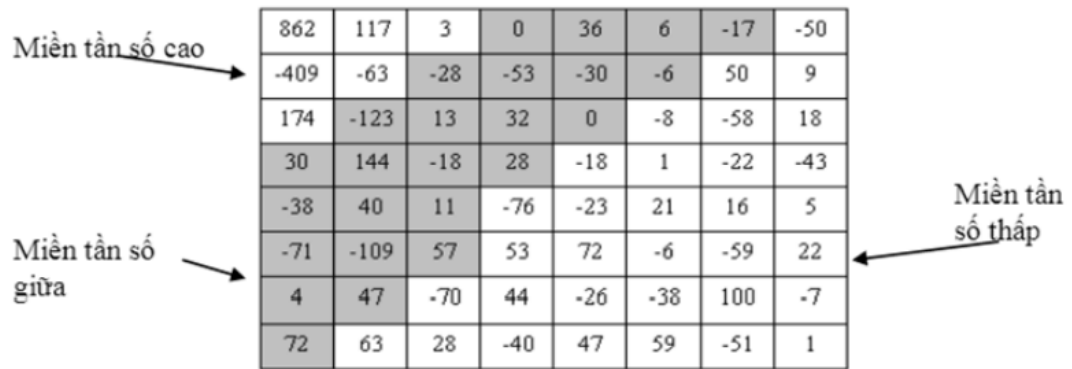
	Low	Horizontal						High
Low	1	2	6	7	15	16	28	29
	3	5	8	14	17	27	30	43
	4	9	13	18	26	31	42	44
	10	12	19	25	32	41	45	54
	11	20	24	33	40	46	53	55
	21	23	34	39	47	52	56	61
	22	35	38	48	51	57	60	62
High	36	37	49	50	58	59	63	64

Hình 2.15. Ví dụ bảng các hệ số DCT (Nguồn: Tài liệu[d])

Theo nguyên lý chung, khi biến đổi chi tiết giữa các điểm ảnh càng lớn theo một hướng nào đó trong khối các điểm ảnh (hướng ngang, hướng thẳng đứng hay theo hướng đường chéo) thì các hệ số biến đổi DCT đương ứng cũng lớn.

Tóm lại, DCT làm giảm độ tương quan không gian của thông tin trong khối ảnh. Điều đó, cho phép biểu diễn thích hợp ở miền DCT do các hệ số DCT có xu hướng có phần dư thừa ít hơn. Hơn nữa, các hệ số DCT chứa thông tin về nội dung tần số không gian của thông tin trong khối. Nhờ các đặc tính tần số không gian của hệ thống nhìn của mắt người, các hệ số DCT có thể được mã hóa phù hợp, chỉ các hệ số DCT quan trọng nhất mới được mã hóa để truyền đi.

Khối hệ số DCT có thể chia thành ba miền: miền tần số thấp, miền tần số giữa và miền tần số cao. Miền tần số thấp chứa các thông tin quan trọng ảnh hưởng đến tri giác. Các thông tin trong miền tần số cao thường không mang tính tri giác cao, khi nén JPEG thì thường loại bỏ thông tin trong miền này.



Hình 2.16. Phân chia 3 miền tần số thấp, giữa, cao của phép biến đổi DCT
(Nguồn: Tài liệu[d])

2.5.3. Mã hóa ảnh dựa trên biến đổi sóng rời rạc (DWT)

Trong phép biến đổi này, Wavelets là các hàm được định nghĩa trong khoảng hữu hạn và có giá trị trung bình bằng 0. Ý tưởng cơ bản của phép biến đổi con sóng con là khai triển hàm $f(t)$ bất kỳ như một xếp chồng của các con sóng con hay các hàm cơ sở. Các hàm cơ sở này có được từ một con sóng con nguyên mẫu được gọi là con sóng mẹ bằng cách lấy tỷ lệ và dịch.

Trong thực tế tính toán, biến đổi con sóng con rời rạc thuận và nghịch (DWT và IDWT) thường được thực hiện bởi phương trình sau:

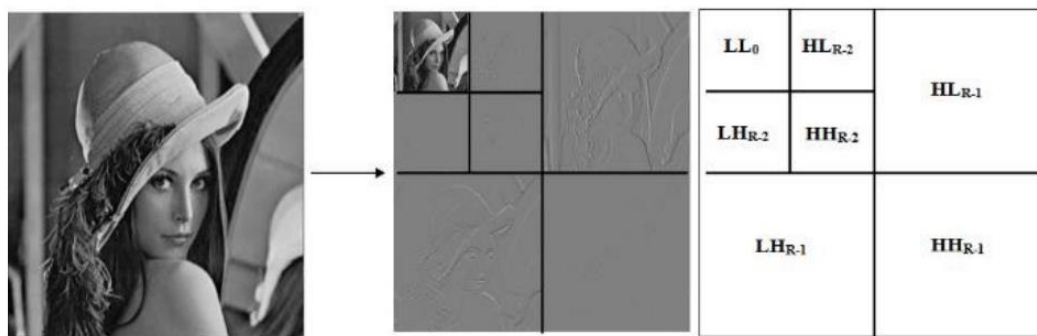
$$\text{Biến đổi thuận: } DWT_f(m, n) = \alpha_0^{\frac{-m}{2}} \int_{-\infty}^{+\infty} f(t) \psi^*(\alpha_0^{-m} t - n b_0) dt$$

$$\text{Biến đổi nghịch: } f(t) = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} \langle \psi_{m,n}, f \rangle \psi_{m,n}(t)$$

Trong đó, $\psi(t)$ là hàm wavelet mẹ. Điều kiện $\psi(t)$ là một hàm thông dải đảm bảo sự tồn tại của biến đổi sóng con ngược. Thông thường, người ta chọn $\alpha_0 = 2$ và $b_0 = 1$.

Trong các thuật toán nghiên cứu có các thông số đáng chú ý sau đây: Tại một miền phân giải cấp 1 thì các hệ số của băng tần xấp xỉ sẽ được gọi là $v1(x, y)$. Các hệ số của băng tần HH1 sẽ được gọi là $f_{1,1}(x, y)$ của LH1 sẽ là $f_{2,1}(x, y)$ và của HL1 là $f_{3,1}(x, y)$.

Với vài thuật toán các hệ số này sẽ được thăm theo đường zig zag. Khi đó, ta sẽ gọi các hệ số này lần lượt theo thứ tự như sau: $v1(i)$, $f_{1,1}(i)$, $f_{2,1}(i)$, $f_{3,1}(i)$.



Hình 2.17. Biến đổi Wavelet và cấu trúc dải thông (Nguồn: Tài liệu[d])

Trong một số trường hợp, sơ đồ dùng biến đổi sóng con đã tỏ ra ưu thế so với biến đổi Fourier rời rạc DFT hay biến đổi cosin rời rạc DCT. Do đặc tính đa phân giải, sơ đồ mã hóa Wavelets đặc biệt thích hợp cho các ứng dụng mà tính vô hướng và suy biến đóng vai trò quan trọng. Minh chứng cho điều này là biến đổi sóng con đã được dùng như một tiêu chuẩn trong nén JPEG2000. Ngoài ra, tính đa phân giải của Wavelets còn hữu ích trong việc phân phối thông điệp vào đối tượng bao phủ trong khi vẫn đảm bảo tính bền vững và chất lượng hiển thị. Do đó, lược đồ thủy văn sử dụng DWT vẫn đảm bảo được tính bền vững của thủy văn sau khi nén có mất mát thông tin theo chuẩn nén JPEG2000.

Phép biến đổi sóng con thực hiện triển khai tần số không gian đa tỷ lệ của một ảnh. Khai triển này tạo ra các hệ số xấp xỉ và các hệ số chi tiết ngang, dọc và chéo. Quá trình khai triển lại tiếp tục với các hệ số xấp xỉ ở mức phân tích cao hơn. Các hệ số xấp xỉ sau cùng chứa thông tin về băng tần thấp nhất trong khi các hệ số chi tiết chứa thông tin về băng tần cao hơn.

CHƯƠNG 3. NGHIÊN CỨU THUẬT TOÁN MÃ HÓA VÀ GIẢI MÃ BẢO VỆ HÌNH ẢNH Y TẾ

3.1. Giới thiệu về hình ảnh y tế

Trong lĩnh vực y tế, hình ảnh đóng một vai trò quan trọng trong việc chẩn đoán, theo dõi và điều trị các vấn đề sức khỏe. Tuy nhiên, hình ảnh y tế thường chứa thông tin nhạy cảm về bệnh nhân và tình trạng sức khỏe của họ. Vì vậy, việc bảo vệ và duy trì tính riêng tư của dữ liệu hình ảnh y tế trở thành một vấn đề quan trọng.

Các hình ảnh y tế thường bao gồm thông tin về các bộ phận cơ thể, bệnh lý, và tình trạng sức khỏe của bệnh nhân. Điều này tạo nên các đặc điểm độc nhất mà có thể dẫn đến việc xác định danh tính và bảo mật cá nhân. Để đảm bảo tính riêng tư và ngăn chặn việc truy cập trái phép, các biện pháp bảo vệ dữ liệu hình ảnh y tế cần được thực hiện.

Một số đặc trưng của hình ảnh y tế bao gồm dữ liệu về cấu trúc cơ thể, bộ phận bị ảnh hưởng bởi bệnh lý, cũng như thông tin về tiến triển của quá trình điều trị. Điều này khiến cho hình ảnh y tế trở thành một tài liệu vô cùng nhạy cảm và đáng được bảo vệ.

Có nhiều mối đe dọa có thể ảnh hưởng đến tính riêng tư của dữ liệu hình ảnh y tế. Một trong những mối đe dọa quan trọng nhất là việc truy cập trái phép, khi người không được ủy quyền có khả năng truy cập vào thông tin nhạy cảm của bệnh nhân. Mối đe dọa khác bao gồm việc thay đổi dữ liệu hình ảnh để tạo ra thông tin sai lệch hoặc gây nhầm lẫn trong việc chẩn đoán và điều trị.

Vai trò của mã hóa trong bảo vệ ảnh y tế là tạo ra một lớp bảo mật bổ sung để đảm bảo tính riêng tư và độ tin cậy của dữ liệu hình ảnh. Các phương pháp mã hóa như biến đổi giá trị, hoán vị pixel và hệ thống hỗn loạn có thể được áp dụng để che giấu thông tin, làm cho dữ liệu trở nên phức tạp để đọc và phục hồi.

Trong ngành y tế, việc bảo vệ hình ảnh không chỉ đảm bảo tính riêng tư của bệnh nhân mà còn là một phần quan trọng của việc duy trì chất lượng dịch vụ y tế. Thông tin trong hình ảnh y tế đóng vai trò quan trọng trong quá trình chẩn đoán, theo dõi và điều trị bệnh. Nếu thông tin này bị rò rỉ hoặc bị sửa đổi một cách trái

phép, có thể gây hại nghiêm trọng cho bệnh nhân và ảnh hưởng đến quyết định y tế. Do đó, việc áp dụng các biện pháp bảo mật, bao gồm cả mã hóa, là cần thiết để đảm bảo tính toàn vẹn và bảo mật của dữ liệu hình ảnh y tế.

3.2. Thuật toán mã hóa bảo vệ hình ảnh y tế

3.2.1. Giới thiệu về các mẫu zigzag, xoay vòng, và hoán vị ngẫu nhiên

Các phương pháp này thường được sử dụng để tái sắp xếp các thông tin hình ảnh thành một cấu trúc không rõ ràng và khó đoán để ngăn chặn việc truy cập và phân tích trái phép.

Các Mẫu Zigzag: Phương pháp này liên quan đến việc xếp các pixel của ảnh theo một mẫu zigzag, tức là từ trái sang phải, từ trên xuống dưới, và ngược lại. Điều này làm cho dữ liệu trở nên phức tạp hơn để theo dõi và phục hồi.

Xoay Vòng: Phương pháp này liên quan đến việc xoay vòng các mẫu pixel theo một kiểu xác định. Sự thay đổi này làm cho việc truy cập dữ liệu trở nên khó khăn hơn và ngăn chặn các cuộc tấn công dựa trên việc phân tích dữ liệu.

Hoán Vị Ngẫu Nhiên: Trong phương pháp này, các pixel trong hình ảnh được hoán vị theo một cách ngẫu nhiên, tạo ra một biểu đồ không thể đoán trước được. Điều này làm cho việc truy cập và phục hồi dữ liệu gốc trở nên rất khó khăn.

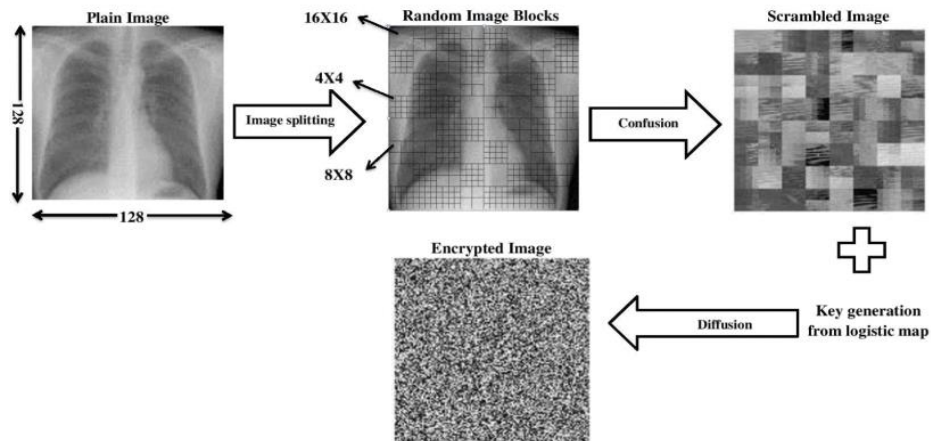
3.2.2. Ánh xạ logistic hỗn loạn

Ánh xạ logistic hỗn loạn (Logistic Map Chaos Mapping) là một phương pháp mã hóa không tuyến tính dựa trên lý thuyết hỗn loạn. Ánh xạ logistic sử dụng một phương trình phi tuyến để biến đổi các giá trị pixel trong hình ảnh. Quá trình này tạo ra sự biến đổi ngẫu nhiên và không thể đoán được của dữ liệu, làm cho việc phục hồi dữ liệu gốc trở nên khó khăn.

Phương pháp này tận dụng tính chất không thể đoán trước của hỗn loạn để bảo vệ dữ liệu hình ảnh y tế khỏi việc truy cập trái phép và phân tích không mong muốn.

Thuật toán mã hóa hình ảnh y tế bao gồm bốn giai đoạn. Trong giai đoạn đầu tiên, thực hiện tách ảnh. Giai đoạn thứ hai thực hiện xáo trộn hình ảnh. Giai đoạn thứ ba trình bày việc tạo khóa dựa trên logistic map. Giai đoạn cuối cùng trình bày

quá trình khuếch tán. Một sơ đồ minh họa về mã hóa hình ảnh y tế được thể hiện trong hình dưới đây.



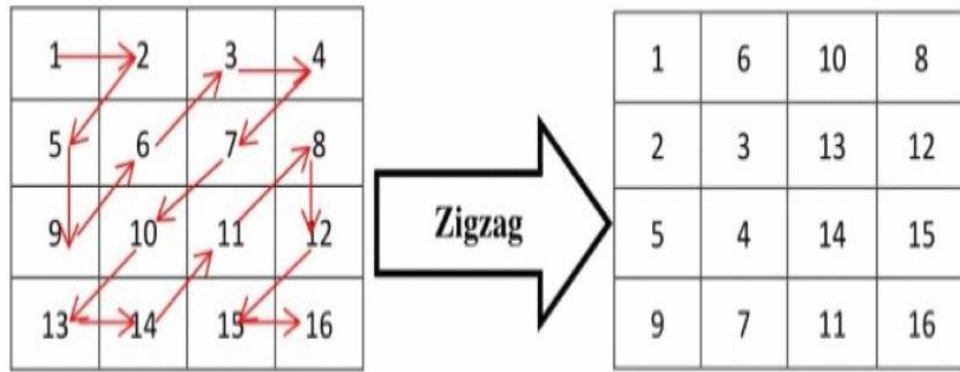
Hình 3.1. Các vòng tiếp theo của thuật toán trộn Fisher – Yates (Nguồn: Tài liệu[e])

3.2.3. Tách hình ảnh gốc

Hình ảnh gốc được chia thành các khối không chồng lấp có cùng kích thước. Thuật toán được trình bày phù hợp với các kích thước khối khác nhau (có thể là 16, 32 và 64,...) và người dùng có thể chọn kích thước khối. Sau đó, mỗi khối được chia nhỏ thành các khối con có kích thước bằng nhau hoặc giữ nguyên mà không bị chia tách. Các khối phụ trong mỗi khối được chọn tùy thuộc vào một số ngẫu nhiên được tạo cho mỗi khối.

Xáo trộn là quá trình thay đổi cách sắp xếp pixel trong ảnh. Trong thuật toán này, thực hiện xáo trộn các khối và khối con như sau:

- Mẫu zigzag được áp dụng cho cả khối ban đầu và khối con, như được mô tả trong 0
- Cả khối không ban đầu (chưa phân tách) và khối con đều xoay 90 độ.
- Vector ngẫu nhiên r được tạo khi kích thước của nó bằng với số lượng khối trong hình ảnh gốc.
- Hoán vị ngẫu nhiên giữa các khối dựa trên vector r để có được hình ảnh xáo trộn.



Hình 3.2. Mẫu zigzag (Nguồn: Tài liệu[e])

3.2.4. Tạo khóa

Khóa được sử dụng trong quá trình khuếch tán được tạo từ logistic map. Logistic map được xác định bởi:

$$Y_{n+1} = aY_n(1 - Y_n)$$

trong đó a là tham số điều kiện với $0 < a \leq 4$, Y_0 là giá trị ban đầu và Y_N là chuỗi đầu ra với $0 < Y_N < 1$. Phép ánh xạ logistic trở nên hỗn loạn (chaotic) khi a nằm trong khoảng $[3.57, 4]$. Hình ảnh bên dưới mô tả biểu đồ phân nhánh (bifurcation diagram) của logistic map. Các bước tạo khóa được định nghĩa như sau:

Tính toán giá trị ban đầu của logistic map dựa vào hình ảnh gốc P bằng phương trình sau đây:

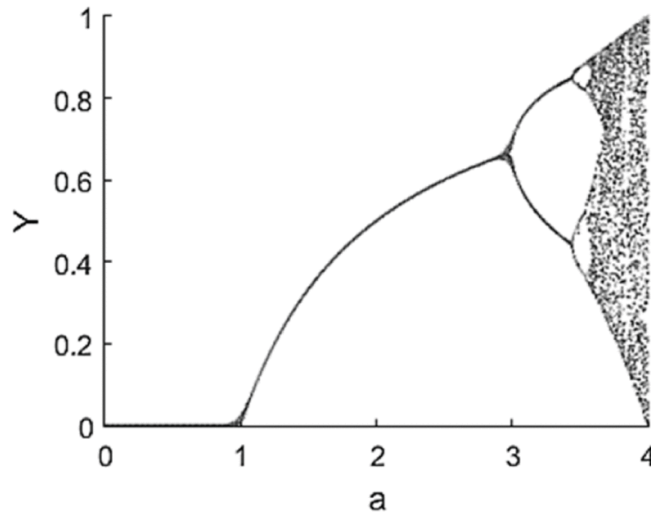
$$Y_0 = \frac{\sum_{i=1}^M \sum_{j=1}^N P(i,j)}{M \times N \times 255}$$

Với các số M và N tương ứng với số hàng và cột trong hình ảnh gốc.

Lặp lại chaotic map $N_0 + MN$ lần, sau đó bỏ qua N_0 phần tử đầu tiên để nhận được một chuỗi mới S có kích thước là MN .

Tính khóa theo công thức sau:

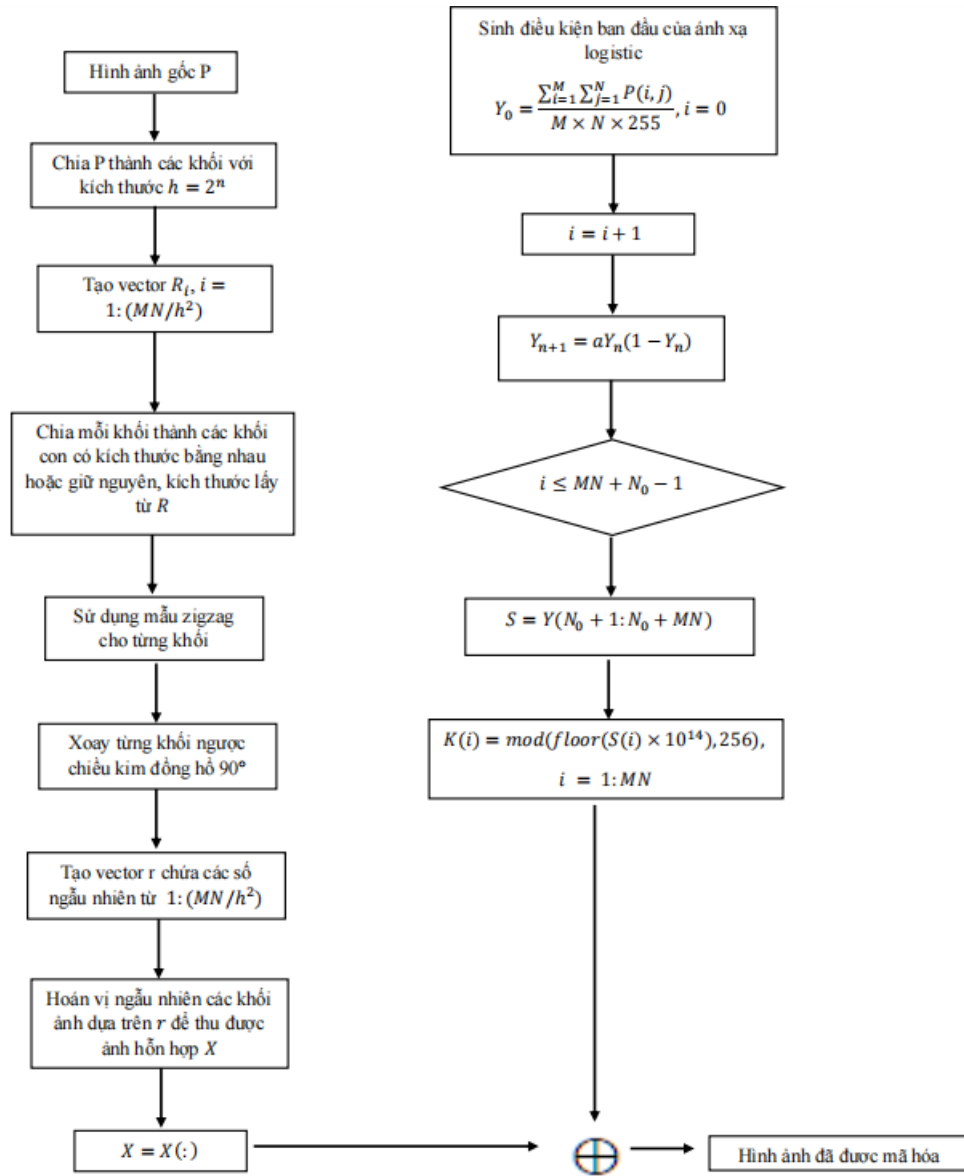
$$K(i) = \text{mod}(\text{floor}(S(i) \times 10^{14}), 256), i = 1:MN$$



Hình 3.3. Sơ đồ phân nhánh của logistic map (Nguồn: Tài liệu[e])

3.2.5. Khuếch tán

Trong quá trình khuếch tán, các giá trị pixel của ảnh được thay đổi tạo ra ảnh nhiễu. Phép XOR bit-wise giữa khóa K và vector hình ảnh đã bị xáo trộn được thực hiện để thu được hình ảnh được mã hóa. Các bước mã hóa chi tiết được trình bày trong Thuật toán 1. Ngoài ra, sơ đồ luồng của thuật toán trong báo cáo được thể hiện trong hình dưới đây.



Hình 3.4. Sơ đồ luồng thuật toán (Nguồn: Tài liệu[e])

Đầu vào: Hình ảnh gốc P với kích thước $M \times N$, tham số a của logistic map và N_0 .

- Chia P thành các khối bằng nhau với kích thước khối $h = 2^n$, trong đó n nhận giá trị từ 4, 5, 6.
- Tạo một số ngẫu nhiên cho mỗi khối dựa trên kích thước khối 2^l trong đó l là một số ngẫu nhiên từ 2 đến n. Đặt các số ngẫu nhiên được tạo trong vector $R_l, i = 1: (MN/h^2)$.
- For $i = 1: MN/h^2$ do

- Chia một khối thành các khối con có cùng kích thước hoặc giữ nguyên mà không chia dựa trên R_i .
- End for
- Thực hiện mẫu Zigzag và xoay vòng góc 90 tương ứng cho cả khối không phân chia và khối đã phân chia.
- Sinh một vector ngẫu nhiên r với kích thước MN/h^2 .
- Hoán vị ngẫu nhiên các khối ảnh dựa trên r để thu được ảnh hỗn hợp X .
- Sinh điều kiện ban đầu của ánh xạ logistic.
- Lặp lại chaotic map $N_0 + MN$ lần, sau đó loại bỏ N_0 phần tử đầu tiên để nhận một chuỗi S mới có kích thước MN .
- For $i = 1: MN$ do
- $K(i) = \text{mod}(\text{floor}(S(i) \times 10^{14}), 256)$
- End for
- Chuyển đổi ma trận X thành vector pixel 1D X' .
- $E = X' \oplus K$
- Chuyển đổi E thành ma trận 2D C .

Đầu ra: hình ảnh được mã hóa C .

3.3. Thuật toán giải mã hình ảnh y tế

Sử dụng khóa ban đầu và đảo ngược các giai đoạn mã hóa, có thể truy xuất hình ảnh gốc. Quá trình giải mã được mô tả như sau:

Áp dụng phép toán XOR bit-wise giữa khóa K và vector hình ảnh mã hóa để thu được hình ảnh xáo trộn.

Đưa mỗi khối về vị trí ban đầu của nó bằng cách sử dụng vector r .

Áp dụng phép toán nghịch đảo của quá trình xoay vòng và mẫu zigzag tương ứng cho các khối chưa được phân tách và các khối con.

3.4. Kết quả mô phỏng trong [1]

Phân tích hệ số tương quan (analysis of correlation coefficient)

Về cơ bản, trong hình ảnh gốc, các pixel liền kề thể hiện mối tương quan cao vì giá trị của nó gần giống hệt nhau. Hiệu quả của thuật toán mã hóa dựa trên việc tạo ra một hình ảnh được mã hóa với mối tương quan thấp giữa các pixel liền kề. Về mặt toán học, hệ số tương quan giữa hai pixel liền kề được xác định bởi các phương trình sau:

$$r_{A,B} = \frac{E\left((A - E(A))(B - E(B))\right)}{\sqrt{D(A)D(B)}}$$

$$E(A) = \frac{1}{s} \sum_{i=1}^s A_i$$

$$D(A) = \frac{1}{s} \sum_{i=1}^s (A_i - E(A))^2$$

Trong đó A và B là các giá trị xám của hai pixel liền kề và s là tổng số cặp được chọn (A, B) . Bảng 4 trình bày các giá trị hệ số tương quan cho cả hình ảnh xám và hình ảnh được mã hóa của chúng theo hướng ngang (H), dọc (V), chéo (D). Tất cả các hình ảnh thử nghiệm có giá trị hệ số tương quan gần bằng một, tuy nhiên, các giá trị hệ số tương quan của hình ảnh được mã hóa gần bằng không. So sánh với các phương pháp khác dựa trên Img1 được liệt kê trong Bảng 5. Thử nghiệm này chứng minh rằng thuật toán đề xuất làm giảm hiệu quả mối tương quan của các pixel liền kề trong hình ảnh được mã hóa.

Test image	Direction	Plain image	Encrypted image
Img2	V	0.9863	-0.0108
	H	0.9727	0.0182
	D	0.9656	0.0165
Img3	V	0.9917	-0.0237
	H	0.9934	0.0059
	D	0.9849	0.0080
Img4	V	0.9730	0.0046
	H	0.9543	-0.0044
	D	0.9414	0.0081
Img5	V	0.9841	0.0063
	H	0.9837	0.0298
	D	0.9708	0.0012

Hình 3.5. Giá trị hệ số tương quan (Nguồn: Tài [e])

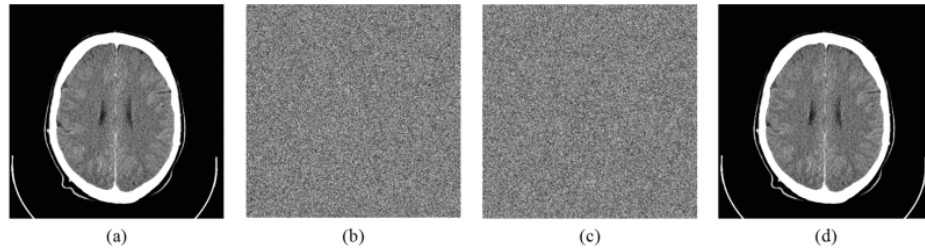
Method	H	V	D
Proposed	-0.0093	0.0025	-0.0024
[23]	-0.0012	0.0099	-0.0032
[24]	0.0027	0.0015	0.0019
[25]	0.0023	-0.0010	0.0009
[26]	0.0944	0.0057	0.0067
[15]	0.0098	-0.0078	0.0181

Hình 3.6. So sánh giá trị hệ số tương quan giữa thuật toán đề xuất và các thuật toán khác. (Nguồn: Tài liệu[e])

Phân tích về độ nhạy khóa

Một thuật toán sẽ dễ bị ảnh hưởng bởi bất kỳ thay đổi nhỏ nào đối với khóa bí mật của nó. Những kẻ tấn công có thể phá vỡ thuật toán mã hóa bằng cách sử dụng một khóa tương tự, vì vậy bất kỳ sự thay đổi nhỏ nào trong khóa sử dụng trong bước giải mã cũng không thể khôi phục lại hình ảnh gốc. Ở đây, tạo ra hai khóa bí mật với chỉ một sự thay đổi nhỏ trong Y_0 bằng cách thay nó thành $Y_0 + 10^{-10}$. Khóa đầu tiên được sử dụng trong bước mã hóa của hình ảnh gốc (hình 2.7a) và kết quả được hiển thị trong hình 2.7b. Khóa thứ hai được sử dụng trong bước giải mã (kết quả thu được trong hình 2.7c). Như có thể thấy, khóa thứ hai không thể khôi phục

lại ảnh gốc. Khi sử dụng khóa đầu tiên để giải mã, thu được thành công hình ảnh gốc như trong hình 2.7d.



Hình 3.7. Độ nhạy của thuật toán của chúng tôi đối với khóa. (a) Hình ảnh gốc. (b) Hình ảnh đã mã hóa của (a) bằng cách sử dụng khóa thứ nhất. (c) Hình ảnh đã giải mã của (b) bằng cách sử dụng khóa thứ hai. (d) Hình ảnh đã giải mã của (b) bằng cách sử dụng khóa thứ nhất. (Nguồn: Tài liệu[e])

Phân tích hiệu quả mã hóa

Tỷ lệ tín hiệu trên nhiễu (PSNR) được sử dụng để đo sự khác biệt giữa hình ảnh gốc và hình ảnh mã hóa và được tính như sau:

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right) (db)$$

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |OI(i,j) - EI(i,j)|^2$$

OI thể hiện cho hình ảnh gốc, EI là hình ảnh đã được mã hóa. Các giá trị PSNR thấp cho thấy sự khác biệt đáng kể giữa hình ảnh gốc và hình ảnh được mã hóa. Hình 2.8 liệt kê các giá trị PSNR cho cá hình ảnh y tế xám khác nhau. Từ kết quả, có thể kết luận rằng thuật toán được đề xuất rất hiệu quả trong việc mã hóa hình ảnh y tế.

Test image	PSNR
Img1	5.1192
Img2	5.8811
Img3	5.6824
Img4	5.2935
Img5	6.0865

Hình 3.8. Phân tích PSNR (Nguồn: Tài liệu[e])

CHƯƠNG 4. THỰC NGHIỆM THUẬT TOÁN MÃ HÓA BẢO VỆ HÌNH ẢNH Y TẾ VÀ GIẢI MÃ

4.1. Mô tả thực nghiệm

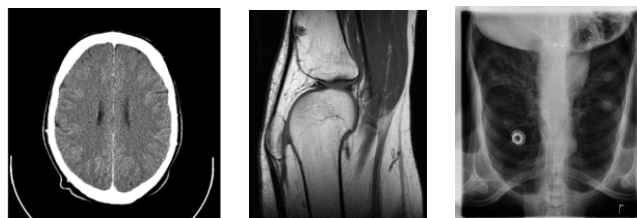
Trong chương này, đề án sẽ tiến hành xây dựng chương trình MATLAB App Designer cho phép mã hóa và giải mã hình ảnh một cách an toàn. Chương trình này cho phép thực hiện các hoạt động sau:

- *Mã hóa hình ảnh*: có thể chọn hình ảnh có định dạng jpg, png, bmp, hoặc gif từ máy tính và mã hóa nó sử dụng thuật toán được đề xuất trong đề án này.
- *Giải mã hình ảnh*: có thể giải mã hình ảnh đã mã hóa trước đó bằng cách sử dụng cùng một thuật toán.

4.1.1. Kịch bản thực nghiệm

- *Kịch bản thực nghiệm*: Xây dựng chương trình mã hóa và giải mã bảo vệ hình ảnh y tế.
- *Mục tiêu*: Xây dựng chương trình hoạt động chính xác và đảm bảo an toàn trong quá trình mã hóa và giải mã hình ảnh. Đồng thời đánh giá thuật toán mã hóa và giải mã.

Tất cả hình ảnh y tế được sử dụng trong phần này được hiển thị trong *0* và có kích thước như 512×512 . Thực thi thuật toán bằng MATLAB(2021b) trên máy tính xách tay CPU Core i5-1035G1 1.00GHz, bộ nhớ 8GB và hệ điều hành Window 11.



(a)img1

(b) img2

(c) img3

Hình 4.1. Hình ảnh thử nghiệm

Các tham số sử dụng trong thuật toán như sau: kích thước của các khối trong quá trình chia nhỏ là 16 ($n = 4$), và đối với ánh xạ logistic có $= 3.9$ và số lần lặp $N0 = 1000$.

4.1.2. Giới thiệu về công cụ MATLAB

MATLAB (MATrix LABoratory) là một môi trường tính toán và lập trình mạnh mẽ phát triển bởi MathWorks. Được sử dụng rộng rãi trong lĩnh vực xử lý hình ảnh, MATLAB cung cấp nhiều tiện ích và khả năng giúp các nhà nghiên cứu, kỹ sư và chuyên gia trong lĩnh vực này thực hiện các phân tích, biến đổi và xử lý hình ảnh một cách hiệu quả.

Trong MATLAB, có một loạt thư viện và hàm xử lý ảnh mạnh mẽ. Nhờ vào chúng, người dùng có khả năng thực hiện các tác vụ như biến đổi, lọc nhiễu, cân bằng màu sắc, trích xuất đặc trưng và nhiều tác vụ xử lý ảnh khác. Giao diện đồ họa thân thiện giúp người dùng thực hiện các tác vụ thông qua các thao tác kéo thả, giúp làm mờ, làm sắc nét, cắt và ghép ảnh trở nên đơn giản và trực quan.

MATLAB cung cấp một ngôn ngữ lập trình mạnh mẽ và linh hoạt cho việc xử lý hình ảnh. Người dùng có thể viết mã lệnh để thực hiện các tác vụ phức tạp hơn và tùy chỉnh các quy trình xử lý ảnh theo nhu cầu cụ thể. Khả năng lập trình hướng đối tượng (OOP) trong xử lý hình ảnh cũng được hỗ trợ.

Không chỉ cung cấp các công cụ và hàm xử lý ảnh, MATLAB còn cho phép người dùng thực hiện các thao tác phức tạp như vẽ đồ thị, biểu đồ và biểu đồ phức tạp, giúp họ hiểu rõ hơn về kết quả xử lý và phân tích ảnh.

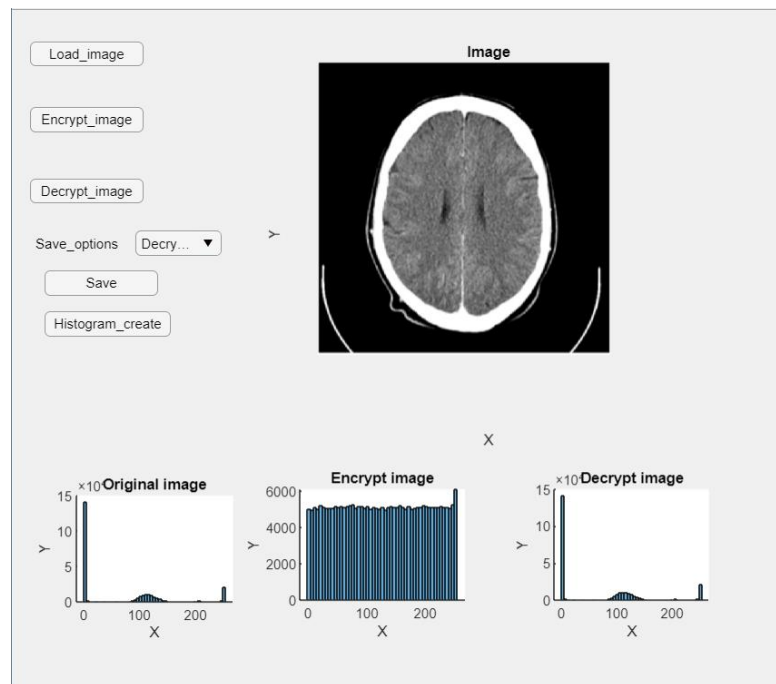
Một điểm mạnh của MATLAB là khả năng tích hợp các công cụ và thư viện xử lý ảnh từ bên ngoài. Người dùng có thể kết hợp MATLAB với các thư viện khác như OpenCV để tận dụng các công nghệ và phương pháp mới nhất trong xử lý ảnh.

Bên cạnh đó, MATLAB hỗ trợ xử lý nhiều loại định dạng hình ảnh khác nhau, bao gồm cả ảnh màu và ảnh đen trắng. Người dùng có thể dễ dàng nhập, xuất và xử lý dữ liệu ảnh từ nhiều nguồn khác nhau.

Trong lĩnh vực xử lý hình ảnh, việc xử lý dữ liệu ảnh có kích thước lớn cũng được hỗ trợ, nhờ vào việc tối ưu hóa và sử dụng các thuật toán hiệu quả.

4.2. Tiến hành thực nghiệm

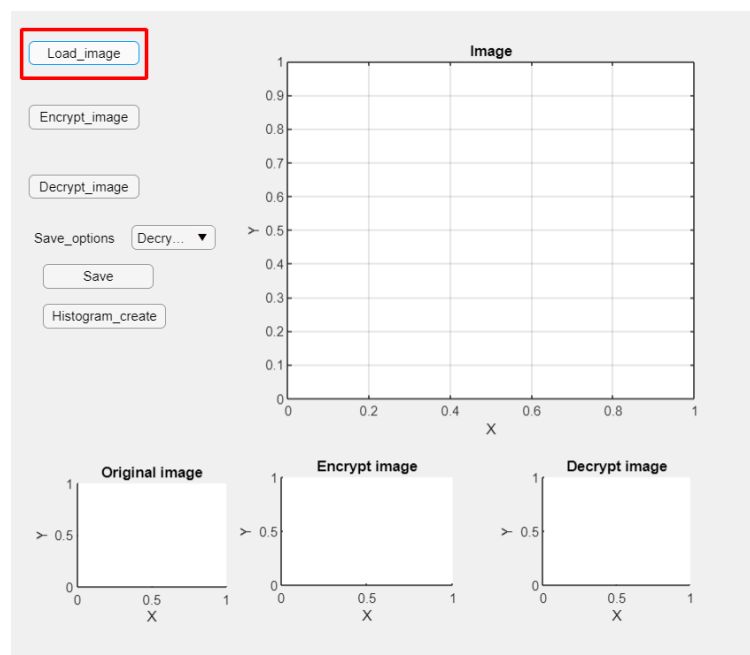
Giao diện chương trình mã hóa và giải mã ảnh:



Hình 4.2. Giao diện chương trình

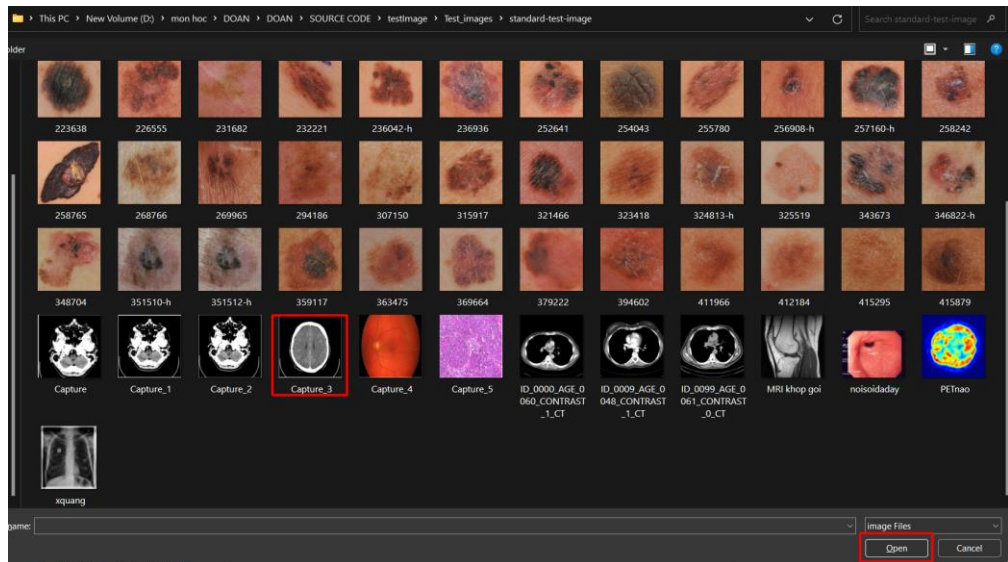
4.2.1. Chương trình mã hóa ảnh

Để chọn hình ảnh cần thực hiện mã hóa, bấm vào nút “Load_image” trên giao diện, trong đó các định dạng thỏa mãn là jpg, png, bmp, gif.



Hình 4.3. Nút lựa chọn hình ảnh đầu vào

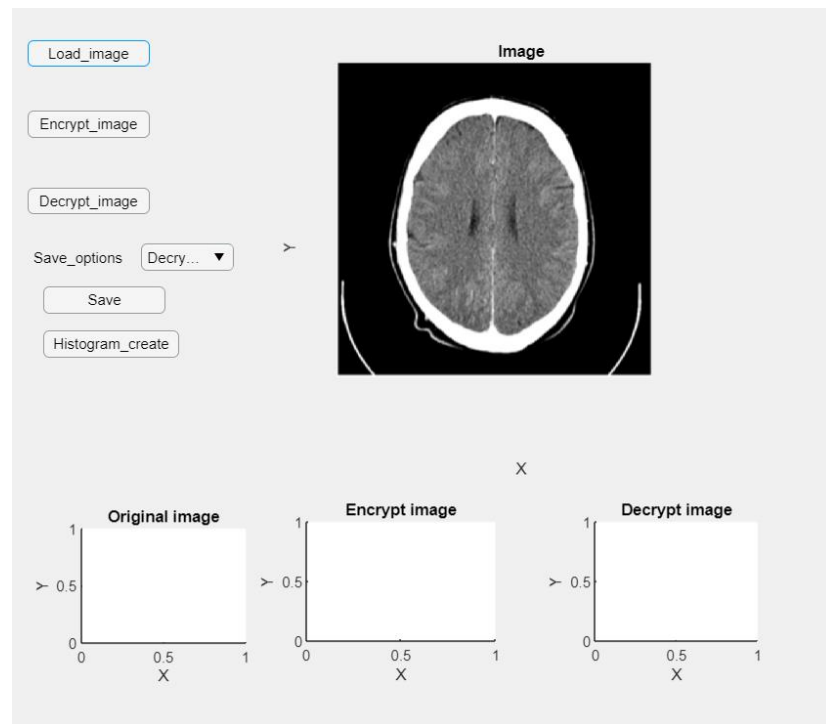
Sau đó cửa sổ “Select an image” sẽ hiện lên, chọn hình ảnh cần mã hóa và bấm “open”.



Hình 4.4. Cửa sổ chọn hình ảnh

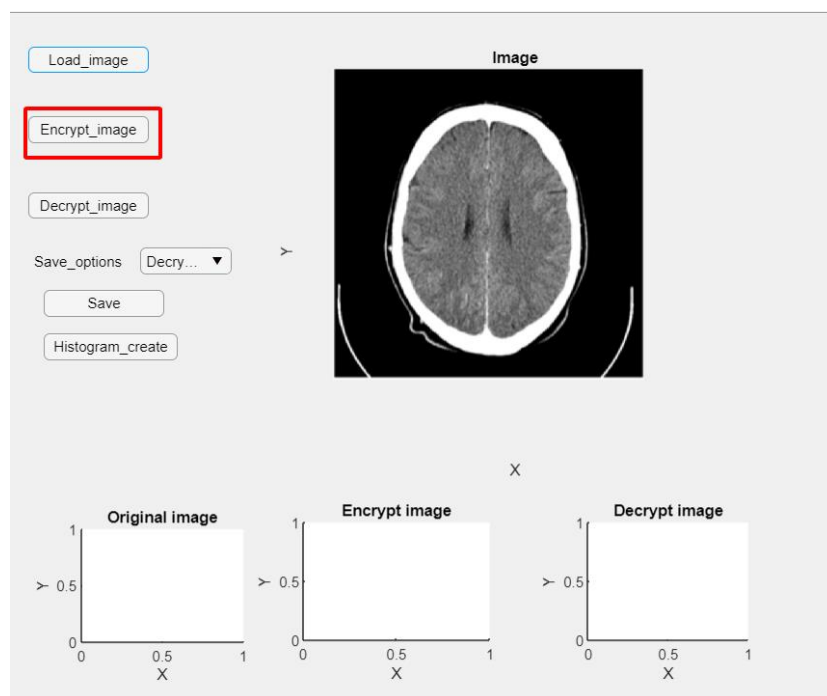
```
[fileName,filePath]=uigetfile({'*.jpg;*.png;*.bmp;*.gif','imageFiles'},'Select an Image');
if isequal(fileName,0)
    return;
end
imagePath=fullfile(filePath,fileName);
app.image=imread(imagePath);
imshow(app.image,'Parent',app.imgAxes)
```

Sau đó hình ảnh sẽ được hiển thị trên giao diện.



Hình 4.5. Hình ảnh sau khi được chọn

Sau khi đã chọn được hình ảnh cần mã hóa, thực hiện mã hóa bằng cách bấm vào nút “Encrypt_image” trên giao diện.



Hình 4.6. Chọn chức năng giải mã hình ảnh

Quá trình mã hóa sẽ thực hiện thông qua các bước sau:

Bước 1: Chia hình ảnh thành các khối và tạo số ngẫu nhiên

Hình ảnh có kích thước 512×512 sẽ được chia thành 1024 khối con ($n=4$). Tại mỗi khối con này, sẽ tạo một số ngẫu nhiên sử dụng để tiếp tục xử lý trên từng khối con.

Bước 2: Xoay vòng, zigzag trên từng khối con

Trên từng khối con đó, tiếp tục phân chia dựa trên các số ngẫu nhiên đã được tạo cho mỗi khối. Thực hiện zigzag và xoay một góc 90 độ theo chiều kim đồng hồ cho các khối này.

Bước 3: Hoán vị các khối con

Các khối con sẽ được hoán vị ngẫu nhiên dựa trên một vector ngẫu nhiên r .

Bước 4: Thực hiện logistic map

Sinh điều kiện ban đầu Y_0 bằng công thức sau:

$$Y_0 = \frac{\sum_{i=1}^M \sum_{j=1}^N P(i, j)}{M \times N \times 255}$$

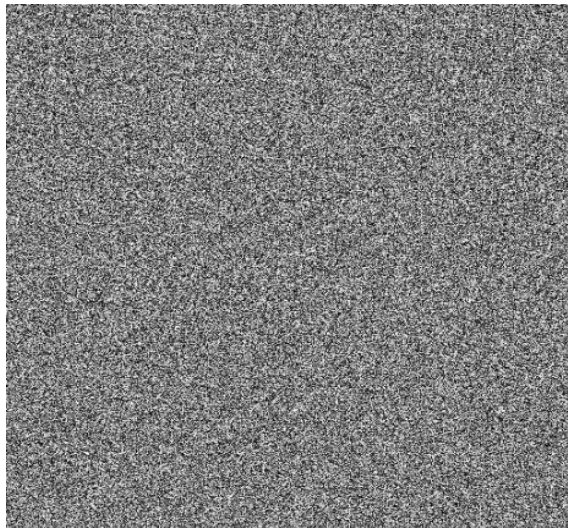
Sau đó thực hiện:

$$Y_{n+1} = aY_n(1 - Y_n)$$

Với $a = 3.9$

Cuối cùng, bitxor hình ảnh đã được hoán vị ở trên với khóa được tạo ra.

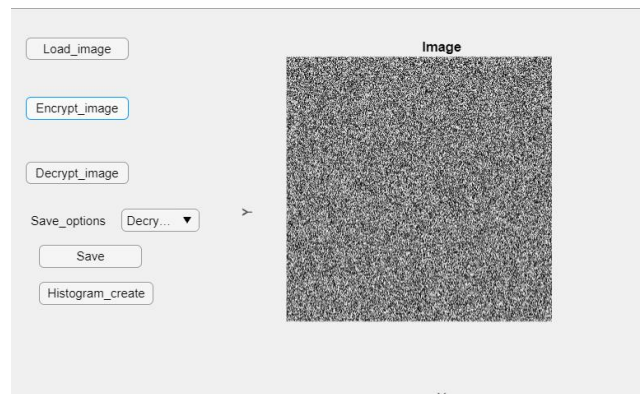
Hình ảnh mã hóa nhận được cho hình ảnh đã chọn như sau:



Hình 4.7. Hình ảnh sau khi mã hóa

4.2.2. Chương trình giải mã ảnh

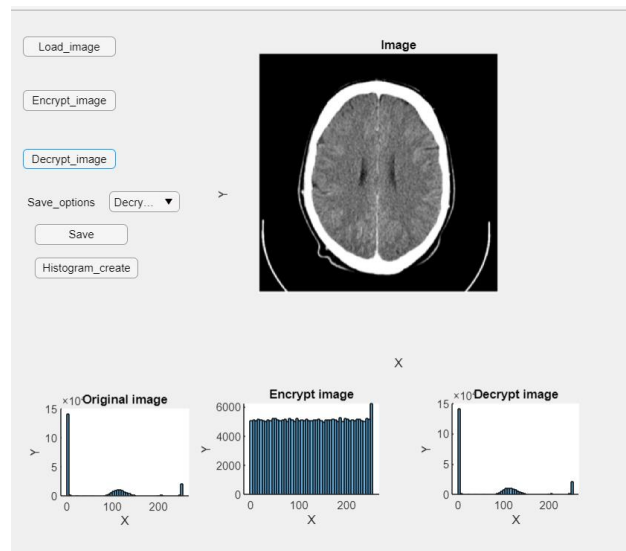
Ảnh mã hóa đầu vào sau bước mã hóa



Hình 4.8. Hình ảnh được hiển thị sau khi mã hóa

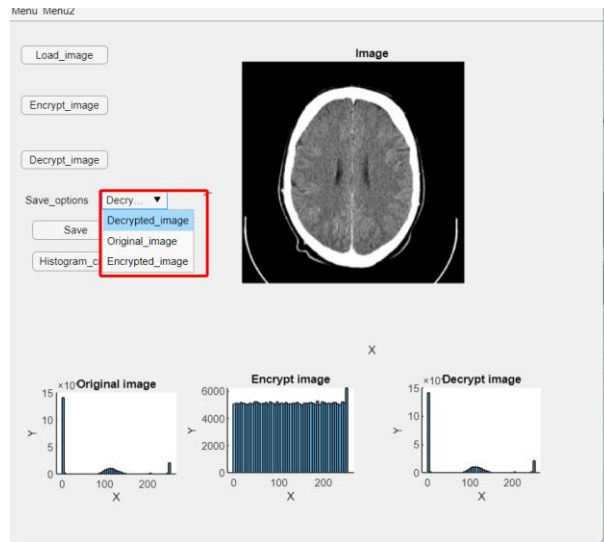
Sau khi đã có được hình ảnh mã hóa, thực hiện giải mã bằng cách bấm vào nút “Decrypt_image” trên giao diện.

Các bước thực hiện giải mã sẽ làm ngược lại với quá trình mã hóa và thu được kết quả như sau:



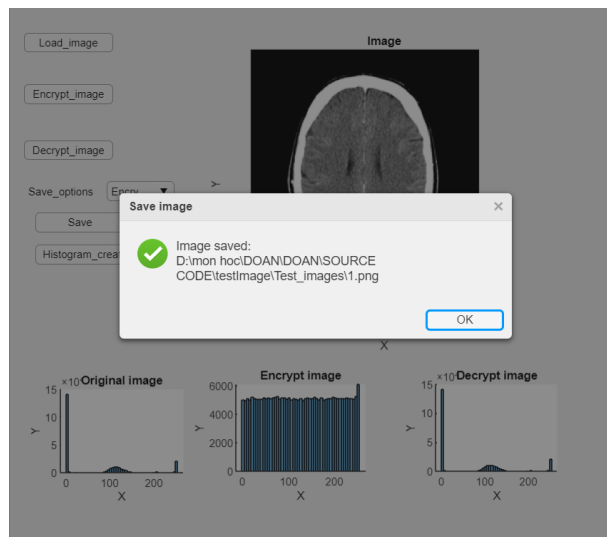
Hình 4.9. Hình ảnh hiển thị sau khi giải mã

Có thể lưu trữ hình ảnh bằng cách chọn “Save_option”, tức là hình ảnh muốn lưu trữ, sau đó bấm “Save”. Một cửa sổ sẽ hiện lên để thực hiện chọn vị trí muốn lưu hình ảnh.



Hình 4.10. Chức năng lưu hình ảnh

Sau khi lưu thành công, chương trình sẽ hiển thị thông báo:



Hình 4.11. Hiển thị sau khi lưu thành công

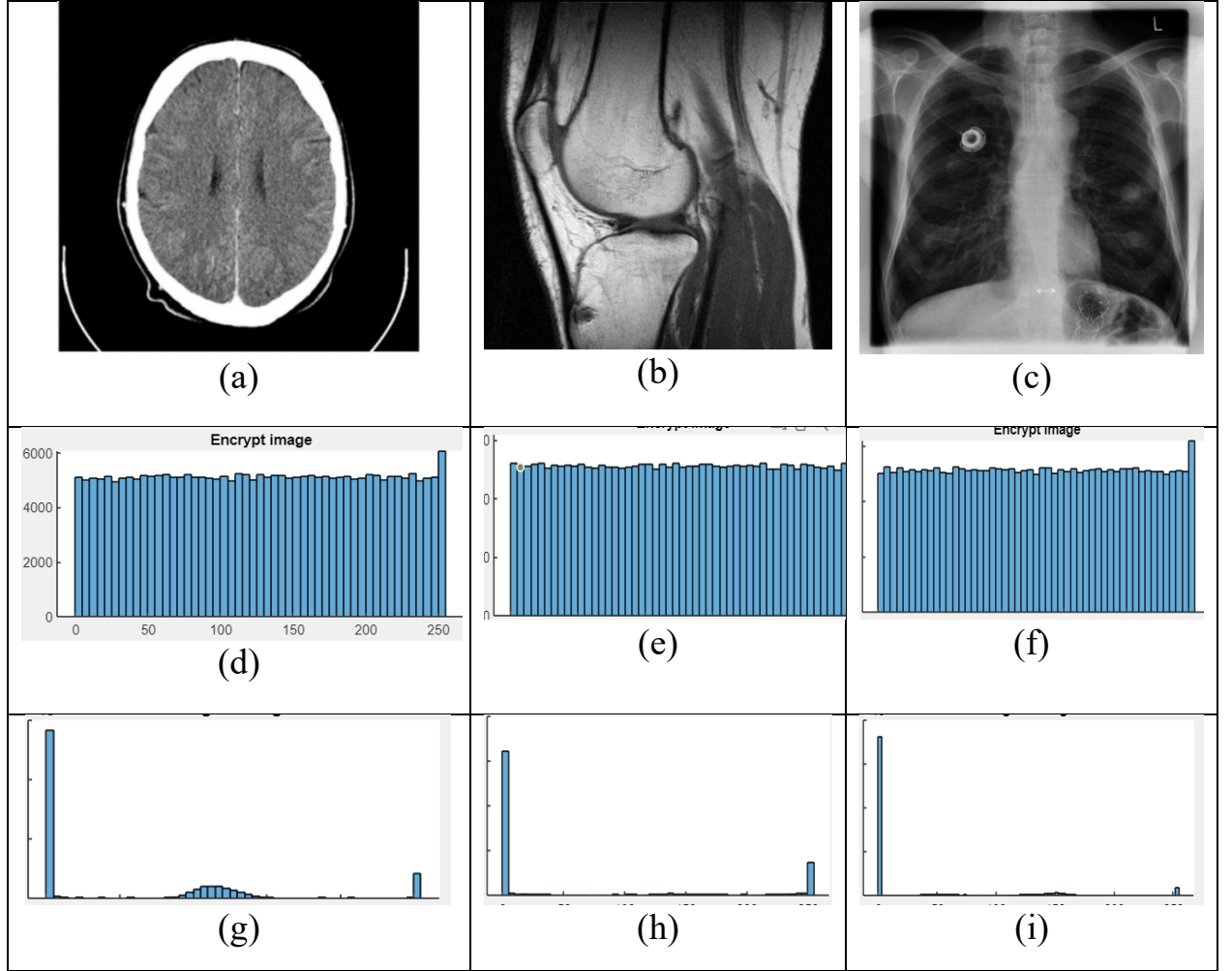
Ngoài ra, có thể hiển thị các histogram của hình ảnh bằng cách bấm chọn “Histogram_create”, sau đó histogram của các hình ảnh gốc, ảnh giải mã và ảnh sau khi mã hóa sẽ được hiển thị trên giao diện.

4.3. Đánh giá kết quả thực nghiệm

4.3.1. Phân tích histogram hình ảnh

Lược đồ histogram biểu thị sự phân bố các pixel trong ảnh. Đối với hình ảnh được mã hóa, lược đồ histogram phải phẳng để ngăn chặn kẻ tấn công đoán được bất kỳ thông tin hình ảnh nào. Ngoài ra, lược đồ của hình ảnh mã hóa và hình ảnh gốc không được giống nhau. 0 thể hiện hiện lược đồ của ba hình ảnh y tế và những hình ảnh được mã hóa của chúng. Như có thể thấy, lược đồ của hình ảnh được mã

hóa bằng thuật toán được đề xuất là đồng nhất và không giống với lược đồ của hình ảnh gốc tương ứng.



Hình 4.12. Phân tích biểu đồ tần suất của hình ảnh xám.

(a) Hình ảnh gốc Img1. (b) Hình ảnh gốc Img2. (c) Hình ảnh gốc Img3. (d) Biểu đồ tần suất của hình ảnh đã mã hóa trong (a). (e) Biểu đồ tần suất của hình ảnh đã mã hóa trong (b). (f) Biểu đồ tần suất của hình ảnh đã mã hóa trong (c). (g) Biểu đồ tần suất của (a). (h) Biểu đồ tần suất của (b). (i) Biểu đồ tần suất của (c).

4.3.2. Phân tích PSNR

Tỷ lệ tín hiệu trên nhiễu (PSNR) được sử dụng để đo sự khác biệt giữa hình ảnh gốc và hình ảnh mã hóa và được tính như sau:

$$PSNR = 10 \times \log_{10} \left(\frac{255^2}{MSE} \right) (db)$$

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |OI(i,j) - EI(i,j)|^2$$

OI thể hiện cho hình ảnh gốc, EI là hình ảnh đã được mã hóa. Các giá trị PSNR thấp cho thấy sự khác biệt đáng kể giữa hình ảnh gốc và hình ảnh được mã hóa. Bảng 3.1 liệt kê các giá trị PSNR cho cá hình ảnh y tế xám khác nhau. Từ kết quả, có thể kết luận rằng thuật toán được đề xuất rất hiệu quả trong việc mã hóa hình ảnh y tế.

Hình ảnh thử nghiệm	PSNR
Img1	6.03
Img2	5.311
Img3	5.584

Bảng 4.1. Phân tích PSNR

4.3.3. Entropy

Mức độ ngẫu nhiên của hình ảnh được đo bằng entropy thông tin. Định nghĩa toán học của entropy được biểu thị bằng:

$$H(m) = \sum_{i=1}^w P(m_i) \log_2 \frac{1}{P(m_i)}$$

Trong đó P(m) là xác suất xuất hiện của m, đối với hình ảnh xám, giá trị tối đa của entropy là 8. Khi giá trị entropy gần 8, sự ngẫu nhiên của các điểm ảnh trong hình ảnh là cao hơn. Trong đồ án này, mã hóa các hình ảnh y tế xám bằng thuật toán được đề xuất và tính toán giá trị entropy của các hình ảnh đã được mã hóa như được liệt kê trong 0. Từ kết quả, có thể thấy rằng tất cả các giá trị entropy đều gần bằng 8, cho thấy tính ngẫu nhiên thực sự của các hình ảnh đã được mã hóa. Hình ảnh đầu tiên (img1) được mã hóa bằng thuật toán được nêu trong báo cáo và các thuật toán khác, như được liệu kê trong 0. Có thể thấy thuật toán được đề xuất trong báo cáo có giá trị entropy cao hơn so với các thuật toán khác trong 0. Từ thí nghiệm này, có thể thấy rằng thuật toán được đề xuất tạo ra các hình ảnh đã được mã hóa với tính ngẫu nhiên cao.

Hình ảnh thử nghiệm	Entropy
Img1	7.927
Img2	7.457
Img3	7.749

Bảng 4.2. Entropy của hình ảnh đã được mã hóa

Phương pháp	Entropy
Phương pháp trong đề án này	7.711
[h]	4.745

Bảng 4.3. Giá trị entropy của phương pháp được đề xuất với các phương pháp khác

4.3.4. Phân tích không gian khóa

Không gian khóa của thuật toán mã hóa hình ảnh tốt ít nhất phải là 2^{100} . Nếu không gian khóa không đủ lớn, thuật toán có thể bị phá vỡ bằng các cuộc tấn công brute-force. Trong thuật toán này, không gian khóa bao gồm điều kiện ban đầu Y_0 , tham số điều khiển α và số lần lặp ban đầu của N_0 của bản đồ hỗn loạn (chaotic map). Ở đây, coi độ chính xác của y_0 là α là 10^{16} và $N_0 = 10^3$. Vì vậy tổng không gian khóa là 10^{35} . Do đó, thuật toán này có thể chống lại các cuộc tấn công brute-force khi không gian khóa đủ lớn.

4.3.5. Phân tích Different attack

Tấn công khác biệt (Differential Attack) phụ thuộc vào việc đoán thông tin về một hình ảnh bằng cách thay đổi nhỏ trong hình ảnh gốc và mã hóa cả hai hình ảnh bằng cùng một thuật toán. So sánh cả hai hình ảnh để phát hiện mối tương quan giữa hình ảnh gốc và hình ảnh đã được mã hóa. Bằng cách sử dụng một thuật toán thực tế, bất kỳ sự thay đổi nhỏ nào trong hình ảnh gốc đều nên tạo ra một hình ảnh đã được mã hóa khác biệt. Để đánh giá hiệu suất của thuật toán, ta sử dụng Tỷ lệ Thay đổi Số điểm ảnh (Number of Pixels Change Rate - NPCR). NPCR được tính như sau:

$$NPCR = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N D(i,j) \times 100(\%)$$

$$D(i,j) = \begin{cases} 0 & \text{if } E_1(i,j) = E_2(i,j), \\ 1 & \text{if } E_1(i,j) \neq E_2(i,j), \end{cases}$$

Các ký hiệu E_1 và E_2 ám chỉ hình ảnh được mã hóa từ hình ảnh gốc và hình ảnh đã được sửa đổi (bằng cách thay đổi một điểm ảnh trong hình gốc). Chiều rộng của hình ảnh là M và chiều cao là N. Trong đồ án này, tính hiệu quả của thuật toán đề xuất trong việc chống lại các cuộc tấn công khác biệt bằng cách ghi lại giá trị NPCR giữa hai hình ảnh đã được mã hóa trong Bảng 3.4. Giá trị lý tưởng của NPCR là 99,6094%. Tất cả các giá trị trong Bảng 3.4 đều gần giống với giá trị lý tưởng. Bảng 3.5 cho thấy sự so sánh giữa thuật toán đề xuất và các thuật toán mã hóa hình ảnh khác. Kết quả cho thấy thuật toán được đề xuất có khả năng chống lại tốt các cuộc tấn công khác biệt.

Hình ảnh thử nghiệm	NPCR
Img1	99.60
Img2	99.67
Img3	99.59

Bảng 4.4. Hiệu suất NPCR

Phương pháp	NPCR
Phương pháp trong đồ án này	99.62
[h]	99.79

Bảng 4.5. So sánh NPCR

KẾT LUẬN

Trong báo cáo này, em đã trình bày thuật toán mã hóa ảnh bảo vệ hình ảnh y tế. Báo cáo cho thấy rằng những mục tiêu đặt ra khi thực hiện đề tài cơ bản đã đạt được.

Trong chương 1, em đã hệ thống lại một số kiến thức căn bản trong mật mã học cũng như trình bày cơ sở lý thuyết để làm nền tảng nghiên cứu thuật toán mã hóa hình ảnh.

Chương 2, em trình bày chi tiết thuật toán mã hóa ảnh bảo vệ hình ảnh y tế.

Chương 3, em đã trình bày kết quả thực nghiệm mã hóa ảnh với thuật toán nghiên cứu. Bên cạnh đó em cũng đưa ra các đánh giá chủ quan của bản thân về hiệu suất thuật toán cũng như kết quả đạt được.

Mặc dù trong quá trình thực hiện em đã rất cố gắng nhưng như một quy luật tất yếu, thuật toán đưa ra vẫn có những hạn chế nhất định bởi có sử dụng khóa bí mật chia sẻ trước lớn. Chính vì thế, vấn đề lưu trữ và chia sẻ khóa bí mật này là một bài toán cần được giải quyết. Việc giải quyết những hạn chế này chính là hướng phát triển của đề tài.

Cuối cùng, em xin chân thành cảm ơn quý thầy cô Khoa Công nghệ thông tin – Trường đại học Đại Nam đã giúp đỡ, tạo điều kiện tốt nhất để em có cơ hội tiếp cận tài liệu nghiên cứu. Đặc biệt, em xin chân thành cảm ơn TS. Bùi Hải Phong, đã giúp đỡ, hướng dẫn em nhiệt tình để em có thể thực hiện đề tài này.

TÀI LIỆU THAM KHẢO

- [a] Sara T. Kamal, Khalid M. Hosny, Taha M. Elgindy, Mohamed M. Darwish, Mostafa M. Fouda. A New Image Encryption Algorithm for Grey and Color Medical Images, 2021.
- [b] Sankar Shanmuganathan, Nagarajan Ls, ZZRD and ZZSW: Novel hybrid scanning paths for squared blocks, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 9, Number 21 (2014) pp. 10567-10582
- [c] Harshvardhan Tiwari, Hamsapriye, N. Satish Kumar. Logistic map based image encryption schemes. International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 23 (2018) pp. 16573-16577, 2021.
- [d] Edgar Chavolla, Daniel Zaldivar, Erik Cuevas and Marco A. Perez. Color Spaces Advantages and Disadvantages in Image Color Clustering Segmentation. Springer International Publishing AG 2018A.E. Hassanien and D.A. Oliva.
- [e] X. Chai, Z. Gan, K. Yuan, Y. Chen, and X. Liu. A novel image encryption scheme based on DNA sequence operations and chaotic systems. Neural Comput. Appl., vol. 31, no. 1, pp. 219–237, Jan. 2019.
- [f] J. Chandrasekaran and S. J. Thiruvengadam. A hybrid chaotic and number theoretic approach for securing DICOM images. Secur. Commun. Netw., vol. 2017, Jan. 2017, Art. no. 6729896.
- [g] X. Wang and C. Liu. A novel and effective image encryption algorithm based on chaos and DNA encoding. Multimedia Tools Appl., vol. 76, no. 5, pp. 6229–6245, Mar. 2017.
- [h] S. Kumar, B. Panna, and R. Kumar. Medical image encryption using fractional discrete cosine transform with chaotic function. Med. Biol. Eng. Comput., vol. 57, no. 11, pp. 2517–2533, 2019.

PHỤ LỤC

Phụ lục 1: Chương trình mã hoá

Chọn ảnh:

```
[fileName,filePath]=uigetfile({'*.jpg;*.png;*.bmp;*.gif','image
Files'},'Select an Image');

if isequal(fileName,0)

    return;

end

imagePath=fullfile(filePath,fileName);

app.image=imread(imagePath);

imshow(app.image,'Parent',app.imgAxes)
```

Zigzag scan:

```
function scan_order=zigzagScan(matrix)

    [rows,cols]=size(matrix);

    % lưu trữ giá trị sau khi thực hiện zigzag
    scan_order=zeros(1,rows*cols);

    rr=1;
    cc=1;

    direction=1;

    for i=1:(rows*cols)

        scan_order(i)=matrix(rr,cc);
```

```

        if direction ==1
            if rr>1 &&cc<cols
                rr=rr-1;
                cc=cc+1;
            elseif cc==cols
                rr=rr+1;
                direction=-1;
            elseif rr==1
                cc=cc+1;
                direction=-1;

            end
        else
            if cc>1&&rr<rows
                rr=rr+1;
                cc=cc-1;
            elseif rr==rows
                cc=cc+1;
                direction=1;
            elseif cc==1
                rr=rr+1;
                direction=1;
            end
        end
    end
end
end
end

```

Chia hình ảnh thành các khối và tạo các số ngẫu nhiên:

```

for ii=1:size(image_new,1)/app.block_size
    for jj=1:size(image_new,2)/app.block_size
        app.R_i(ii,jj)=2^randi([2,4]);
        %block=image((i-1)*block_size+1:i*block_size,(j-
1)*block_size+1:j*block_size);
        block_test{ii,jj}=image_new((ii-
1)*app.block_size+1:ii*app.block_size,(jj-
1)*app.block_size+1:jj*app.block_size);
    end
end

```

Logistic map:

```

%generate điều kiện ban đầu của logistic map
Y_0=
sum(sum(sum(image_new,3)))/(size(image_new,1)*size(image_new,2)*255);

N_0=1000;
a=3.9;
MN=size(image_new,1)*size(image_new,2);
% gán Y_0 cho phân tử đầu tiên của dãy Y
Y(1)=Y_0;
for i=2:MN+N_0
    Y(i)=a*Y(i-1)*(1-Y(i-1));
end
s=Y(N_0+1:N_0+MN);
for i=1:length(s)
    app.K(i)=mod(floor(s(i)*10^14),256);
end

```

Phụ lục 2: Chương trình giải mã

Inverse zigzag scan:

```
[rows,cols]=size(scan_matrix);

matrix=zeros(rows,cols);

r1=1;
c1=1;
direction1=1;

for i1=1:(rows*cols)
    matrix(r1,c1)=scan_matrix(i1);

    if direction1==1
        if r1>1&& c1<cols
            r1=r1-1;
            c1=c1+1;
        elseif c1==cols
            r1=r1+1;
            direction1=-1;
        elseif r1==1
            c1=c1+1;
            direction1=-1;
        end
    else
        if c1>1&& r1<rows
            r1=r1+1;
            c1=c1-1;
        elseif r1==rows
            c1=c1+1;
            direction1=1;
        end
    end
end
```

```

        c1=c1+1;
        direction1=1;
    elseif c1==1
        r1=r1+1;
        direction1=1;
    end
end
end
end
end

```

Giải mã:

```

        %block_size=max(max(app.R_i));

        encrypted_image_2D=app.encrypted_image;

        % chuyển hình ảnh từ 2D sang 1D
        encrypted_image_1D=reshape(encrypted_image_2D,size(app.K));

        %thực hiện xor bitwise bằng khóa để lấy lại hình ảnh được
        %hoán vị

        permuted_image_1D=bitxor(encrypted_image_1D,app.K);

        %show the scrambled
        %figure;

        %imshow(reshape(permuted_image_1D,size(encrypted_image_2D)))
    end
end
end
end

```

```

%định hình lại các hình ảnh đã hoán vị thành các khối

permuted_image_16=reshape(permuted_image_1D,app.block_size,size(perm
    utated_image_1D,2)/app.block_size);

%blocks_permuted=mat2cell(permuted_image_16,block_size,block_size*on
    es(1,size(permuted_image_16,2)/block_size));
    for i=1:size(permuted_image_16,2)/app.block_size

        blocks_permuted{i}=permuted_image_16(:,(i-
            1)*app.block_size+1:i*app.block_size);

        end

        %get back the original blocks image
        original_blocks=cell(size(blocks_permuted));

        for i=1:length(app.r)

            index=app.r(i);
            original_blocks(index)=blocks_permuted(i);

            end

        reshaped_blocks=reshape(original_blocks,size(encrypted_image_2D,1)/app.bloc
            k_size,size(encrypted_image_2D,2)/app.block_size);

        %get subblocks using R_i;
        %figure;
        %imshow(imrotate(imrotate(cell2mat(reshaped_blocks),90),-90))

```

```

        blocks_retrieved=cell(size(reshaped_blocks));
        for i = 1:size(reshaped_blocks, 1)
            for j = 1:size(reshaped_blocks, 2)
                subblock_size = app.R_i(i, j);

                if subblock_size ~= app.block_size
                    bb = reshaped_blocks{i, j};
                    retrieved_subblocks = cell(app.block_size / subblock_size, app.block_size /
                    subblock_size);
                    for x = 1:size(retrieved_subblocks, 1)
                        for y = 1:size(retrieved_subblocks, 2)
                            subblock_retrieved = bb((x - 1) * subblock_size + 1:x * subblock_size, (y - 1) *
                            subblock_size + 1:y * subblock_size);
                            subblock_retrieved = imrotate(subblock_retrieved, -90);
                            subblock_retrieved = inverseZigzagScan(subblock_retrieved);
                            retrieved_subblocks{x, y} = subblock_retrieved;
                        end
                    end
                    block_retrieved = cell2mat(retrieved_subblocks);
                else
                    cc = imrotate(reshaped_blocks{i, j}, -90);
                    block_retrieved = inverseZigzagScan(cc);
                end
                blocks_retrieved{i, j} = uint8(block_retrieved);
            end
        end

        app.decrypted_image=(cell2mat(blocks_retrieved));

```



```
imshow(app.decrypted_image,'Parent',app.imgAxes)
```

Phụ lục 3: Đánh giá

Entropy:

```
% Chuẩn hóa histogram để có xác suất
probability = histogram / sum(histogram);

% Loại bỏ các giá trị có xác suất 0
probability = probability(probability > 0);

% Tính entropy thông tin
info_entropy = -sum(probability .* log2(probability));
```

PNCR:

```
different_pixels = sum(image1(:) ~= image2(:));
total_pixels = numel(image1);
npcr = different_pixels / total_pixels * 100;
```

UACI:

```
intensity_difference = abs(double(image1) - double(image2));
uaci = mean(intensity_difference(:)) / 255 * 100;
```

PSNR:

```
% Tính toán MSE (Mean Squared Error)
mse = sum((original_image(:) - encoded_image(:)).^2) /
      numel(original_image);

% Tính toán PSNR (Peak Signal-to-Noise Ratio)
psnr_value = 10 * log10((255^2) / mse);
```