BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

ISO 9001:2000

ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ THÔNG TIN

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

TÊN ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP

KỸ THUẬT GIẤU TIN THUẬN NGHỊCH SỬ DỤNG THUẬT TOÁN MAXMIN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

HẢI PHÒNG - 2013

NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

Sinh viên: Cao Lê Huân.

Mã SV: 121494.

Lớp: CT1201.

Ngành: Công nghệ Thông tin

Tên đề tài: **KỸ THUẬT GIẤU TIN THUẬN NGHỊCH SỬ DỤNG THUẬT TOÁN MAXMIN**

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Nội dung

- Tổng quan về giấu tin trong ảnh số,
- Thuật toán giấu tin Maxmin
- Tìm hiểu kỹ thuật giấu thuật nghịch trên miền dữ liệu ảnh sử dụng thuật toán Maxmin
- Cài đặt, thử nghiệm chương trình

b. Các yêu cầu cần giải quyết

a) Lý thuyết

- Hiểu được cấu trúc cơ bản của ảnh Bitmap, một số khái niệm cơ bản về xử lý ảnh.
- Nắm được tổng quan về kỹ thuật giấu tin trong ảnh.
- Hiểu và nắm rõ kỹ thuật giấu ảnh màu trong ảnh.

b) Thực nghiệm (chương trình)

- Cài đặt được kỹ thuật giấu bằng Matlab, thử nghiệm trên một tập ảnh để có thể đánh giá độ trực quan của ảnh sau khi giấu tin bằng PSNR, từ đó đưa ra nhận xét về kỹ thuật giấu áp dụng cho tập ảnh thử nghiệm.

2. Các số liệu cần thiết để thiết kế, tính toán.

- Tập ảnh để thử nghiệm.

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Người hướng dẫn thứ nhất:

Họ và tên: Hồ Thị Hương Thơm

Học hàm, học vị: Tiến Sĩ

Cơ quan công tác: Trường Đại Học Dân Lập Hải Phòng

Nội dung hướng dẫn:

Người hướng dẫn thứ hai:

Họ và tên: ……

Học hàm, học vị: ……

Cơ quan công tác: ……

Nội dung hướng dẫn: ……..

……

……

……

……

……...

Đề tài tốt nghiệp được giao ngày tháng năm 2013

Yêu cầu phải hoàn thành trước ngày tháng năm 2013

Đã nhận nhiệm vụ: Đ.T.T.N	Đã nhận nhiệm vụ: Đ.T.T.N
Sinh viên	Cán bộ hướng dẫn Đ.T.T.N

TS. Hồ Thị Hương Thơm

Hải Phòng, ngày .....tháng.....năm 2013

HIỆU TRƯỞNG

GS.TS.NGƯT Trần Hữu Nghị

PHẦN NHẬN XÉT TÓM TẮT CỦA CÁN BỘ HƯỚNG DẪN

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp:

.....
.....
.....
.....
.....
.....

2. Đánh giá chất lượng của đề tài tốt nghiệp (so với nội dung yêu cầu đã đề ra trong nhiệm vụ đề tài tốt nghiệp)

.....
.....
.....
.....
.....
.....
.....
.....

3. Cho điểm của cán bộ hướng dẫn:

(Điểm ghi bằng số và chữ)

.....
.....
.....

Ngày.....tháng.....năm 2013

Cán bộ hướng dẫn chính

(Ký, ghi rõ họ tên)

PHẦN NHẬN XÉT ĐÁNH GIÁ CỦA CÁN BỘ CHẤM PHẢN BIỆN ĐỀ TÀI TỐT NGHIỆP

1. Đánh giá chất lượng đề tài tốt nghiệp (về các mặt như cơ sở lý luận, thuyết minh chương trình, giá trị thực tế, ...)

Cho điểm của cán bộ phản biện

(*Điểm ghi bằng số và chữ*)

.....

.....

.....

Ngày.....tháng.....năm 2013

Cán bộ chấm phản biện

(Ký, ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN

Trước hết em xin bày tỏ lòng biết ơn sâu sắc nhất tới cô giáo hướng dẫn Tiến sĩ Hồ Thị Hương Thơm đã tận tình giúp đỡ em rất nhiều trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành báo cáo tốt nghiệp.

Em xin chân thành cảm ơn các thầy cô trong bộ môn tin học – trường DHDL Hải Phòng cũng như các thầy cô trong trường đã trang bị cho em những kiến thức cơ bản cần thiết để em có thể hoàn thành báo cáo.

Xin gửi lời cảm ơn đến bạn bè những người luôn bên em đã động viên và tạo điều kiện thuận lợi cho em, tận tình giúp đỡ chỉ bảo em những gì em còn thiếu sót trong quá trình làm báo cáo tốt nghiệp.

Cuối cùng em xin bày tỏ lòng biết ơn sâu sắc tới những người thân trong gia đình đã giành cho em sự quan tâm đặc biệt và luôn động viên em.

Vì thời gian có hạn, trình độ hiểu biết của bản thân còn nhiều hạn chế. Cho nên trong đồ án không tránh khỏi những thiếu sót, em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy cô giáo cũng như các bạn bè để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

Hải phòng, ngày… tháng…năm 2013

Sinh viên thực hiện

DANH MỤC HÌNH

DANH MỤC BẢNG

LỜI MỞ ĐẦU

Cuộc cách mạng thông tin kỹ thuật số đã đem lại những thay đổi sâu sắc trong xã hội và trong cuộc sống của chúng ta. Những thuận lợi mà thông tin kỹ thuật số mang lại cũng sinh ra những thách thức và cơ hội mới cho quá trình đổi mới. Với việc sử dụng mạng internet toàn cầu để thông tin, liên lạc ngày càng tăng trong mọi lĩnh vực chính trị, quân sự, quốc phòng, kinh tế, thương mại… Vấn đề được đặt ra đó là sự an toàn của dữ liệu. Một công nghệ phần nào giải quyết được vấn đề trên là giấu tin mật, nó cho phép giấu thông tin mật vào trong các nguồn thông tin khác, làm ẩn đi sự tồn tại của thông mật. Trong đồ án này em xin trình bày một kỹ thuật giấu tin đó là "**Kỹ thuật giấu tin thuận nghịch sử dụng thuật toán MAXMIN**", gồm các chương sau:

Chương 1. Tổng quan về kỹ thuật giấu tin: Khái niệm giấu tin, mục đích của giấu tin, cấu trúc ảnh bitmap, đánh giá chất lượng ảnh bằng PSNR.

Chương 2. Kỹ thuật giấu tin thuận nghịch trong ảnh: Giới thiệu về kỹ thuật giấu tin thuận nghịch sử dụng thuật toán MAXMIN, trình bày một số kỹ thuật giấu tin thuận nghịch, đưa ra thuật toán.

Chương 3. Cài đặt thử nghiệm: Trình bày một số giao diện của chương trình và thử nghiệm kỹ thuật giấu tin thuận nghịch sử dụng thuật toán MAXMIN, đưa ra nhận xét đánh giá.

Chương 1. TỔNG QUAN VỀ KỸ THUẬT GIẤU TIN

1.1 TỔNG QUAN VỀ KỸ THUẬT GIẤU TIN

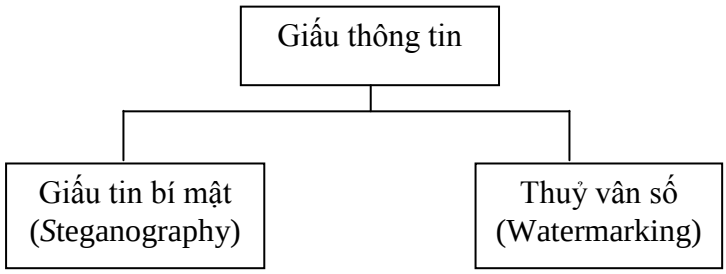
1.1.1 Định nghĩa kỹ thuật giấu tin

Giấu thông tin là một kỹ thuật nhúng (giấu) một lượng thông tin số nào đó vào trong một đối tượng dữ liệu số khác (giấu thông tin chỉ mang tính quy ước không phải là một hành động cụ thể).

1.1.2 Phân loại giấu tin

Có thể phân loại kỹ thuật giấu tin làm hai hướng:

- Giấu tin mật (Steganography).
- Thủy vân số (Watermarking).



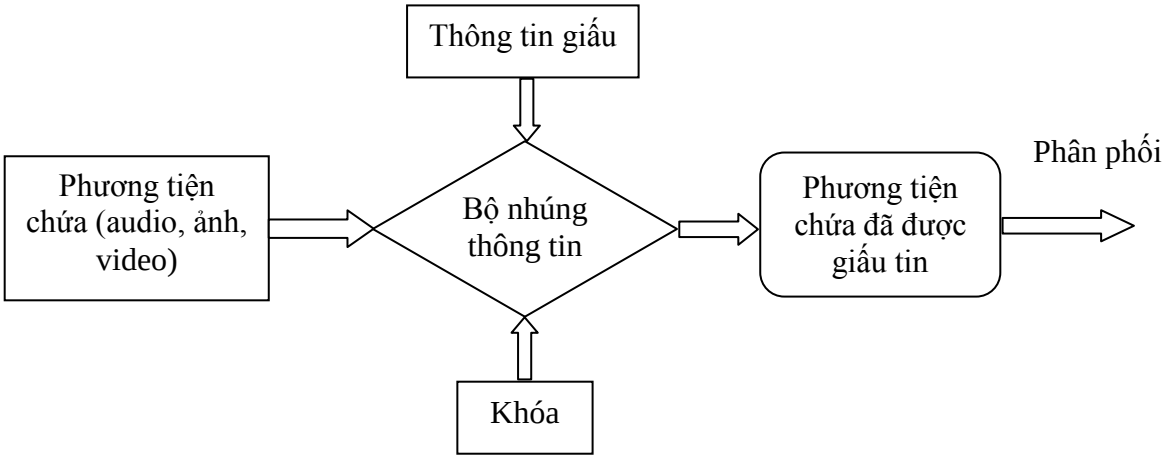
Hình 1.1. *Hai lĩnh vực chính của kỹ thuật giấu thông tin*

Kỹ thuật giấu thông tin bí mật (*Steganography*): với mục đích đảm bảo an toàn và bảo mật thông tin tập trung vào các kỹ thuật giấu tin để có thể giấu được nhiều thông tin nhất. Thông tin mật được giấu một cách vô hình trong một đối tượng khác sao cho người khác khó phát hiện được.

Kỹ thuật giấu thông tin theo kiểu đánh dấu – thủy vân (*watermarking*) với mục đích để bảo vệ bản quyền chính đối tượng dùng để chứa thông tin, thường tập trung đảm bảo một số các yêu cầu như đảm bảo tính bền vững… Đây là ứng dụng cơ bản nhất của kỹ thuật thuỷ vân số.

1.1.3 Mô hình kỹ thuật giấu thông tin cơ bản

Giấu thông tin vào phương tiện chứa và tách lấy thông tin là 2 quá trình trái ngược nhau và có thể mô tả qua sơ đồ của hệ thống như Hình 1.2:



Hình 1.2 *Lược đồ chung cho quá trình giấu tin*

Thông tin cần giấu tuỳ theo mục đích của người sử dụng, nó có thể là thông tin mật (với các tin bí mật) hay các logo, hình ảnh bản quyền.

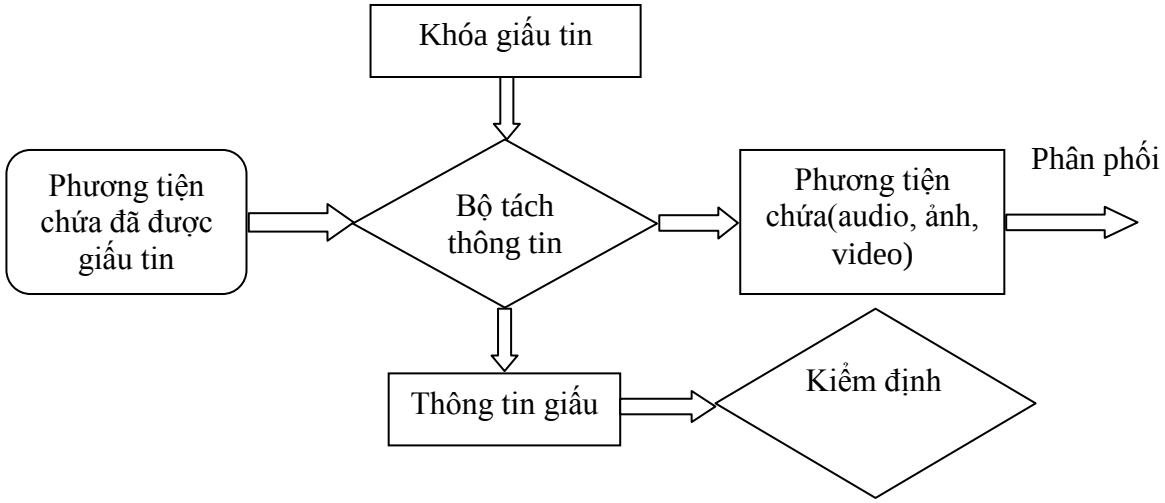
Phương tiện chứa: các *file* ảnh, *text, audio*… là môi trường để nhúng tin.

Bộ nhúng thông tin: là những chương trình thực hiện việc giấu tin

Đầu ra: là các phương tiện chứa đã có tin giấu trong đó

Tách thông tin từ các *phương tiện chứa* diễn ra theo quy trình ngược lại với đầu ra là các thông tin đã được giấu vào *phương tiện chứa*. *Phương tiện chứa* sau khi tách lấy thông tin có thể được sử dụng, quản lý theo những yêu cầu khác nhau.

1.1.4 Mô hình kỹ thuật tách thông tin cơ bản



Hình 1.3 *Lược đồ chung cho quá trình tách thông tin*

Hình 1.3 chỉ ra các công việc giải mã thông tin đã giấu. Sau khi nhận được đối tượng *phương tiện chứa* có giấu thông tin, quá trình giải mã được thực hiện thông qua một bộ giải mã tương ứng với bộ nhúng thông tin cùng với khoá của quá

trình nhúng. Kết quả thu được gồm *phương tiện chứa* gốc và thông tin đã giấu. Bước tiếp theo thông tin đã giấu sẽ được xử lý kiểm định so sánh với thông tin ban đầu.

1.1.5 Yêu cầu thiết yếu đối với một hệ thống giấu tin

Có 3 yêu cầu thiết yếu đối với một hệ thống giấu tin:

- Tính vô hình: là một trong 3 yêu cầu của bất kì 1 hệ giấu tin nào.
- Tính bền vững: là yêu cầu thứ 2 của một hệ giấu tin. Tính bền vững là nói đến khả năng chịu được các thao tác biến đổi nào đó trên phương tiện nhúng và các cuộc tấn công có chủ đích.
- Khả năng nhúng: là yêu cầu thứ 3 của một hệ giấu tin. Khả năng nhúng chính là số lượng thông tin nhúng được nhúng trong phương tiện chứa.

1.1.6 Môi trường giấu tin

a. Giấu tin trong ảnh

- Giấu tin trong ảnh hiện đang rất được quan tâm. Nó đóng vai trò hết sức quan trọng trong hầu hết các ứng dụng bảo vệ an toàn thông tin như: nhận thực thông tin, xác định xuyên tạc thông tin, bảo vệ bản quyền tác giả…
- Một đặc điểm của giấu thông tin trong ảnh nữa đó là thông tin được giấu một cách vô hình, nó như là cách truyền thông tin mật cho nhau mà người khác không thể biết được bởi sau khi giấu thông tin chất lượng ảnh gần như không thay đổi đặc biệt đối với ảnh màu hay ảnh xám.

b. Giấu tin trong audio

- Khác với kỹ thuật giấu thông tin trong ảnh: phụ thuộc vào hệ thống thị giác của con người – HSV (*Human Vision System*), kỹ thuật giấu thông tin trong *audio* lại phụ thuộc vào hệ thống thính giác HAS (*Human Auditory System*). Bởi vì tai con người rất kém trong việc phát hiện sự khác biệt giữa các giải tần và công suất, có nghĩa là các âm thanh to, cao tần có thể che giấu đi được các âm thanh nhỏ, thấp một cách dễ dàng.

- Yêu cầu cơ bản và quan trọng nhất của giấu tin trong *audio* là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không làm ảnh hưởng đến chất lượng của dữ liệu.

c. Giấu tin trong video

- Cũng giống như giấu thông tin trong ảnh hay trong *audio*, giấu tin trong *video* cũng được quan tâm và được phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, xác thực thông tin, bản quyền tác giả…
- Một phương pháp giấu tin trong *video* được đưa ra bởi *Cox* là phương pháp phân bố đều. Ý tưởng cơ bản của phương pháp là phân phối thông tin giấu dàn trải theo tần số của dữ liệu gốc.

d. Giấu thông tin trong văn bản dạng text

- Giấu tin trong văn bản dạng *text* khó thực hiện hơn do có ít các thông tin dư thừa, để làm được điều này người ta phải khéo léo khai thác các dư thừa tự nhiên của ngôn ngữ. Một cách khác là tận dụng các định dạng văn bản (mã hoá thông tin vào khoảng cách giữa các từ hay các dòng văn bản) => Kỹ thuật giấu tin đang được áp dụng cho nhiều loại đối tượng chứ không riêng dữ liệu đa phương tiện như ảnh, *audio, video.*

1.1.7 Một số đặc điểm của việc giấu tin trên ảnh

1.1.7.1 Tính vô hình của thông tin

Khái niệm này dựa trên đặc điểm của hệ thống thị giác của con người. Thông tin nhúng là không tri giác được nếu một người với thị giác bình thường không phân biệt được ảnh môi trường và ảnh kết quả (tức là không phân biệt được ảnh trược và sau khi giấu thông tin). Trong khi giấu thông tin mật *(Steganography)* yêu cầu tính vô hình của thông tin ở mức độ cao thì giấu thông tin theo kiểu đánh dấu – thủy vân (*watermarking*) lại chỉ yêu cầu ở một cấp độ nhất định. Chẳng hạn như người ta áp dụng giấu thông tin theo kiểu đánh dấu – thủy vân (*watermarking*) cho việc gắn một biểu tượng mờ vào một chương trình truyền hình để bảo vệ bản quyền.

1.1.7.2 Khả năngnhúng tin

Lượng thông tin giấu so với kích thước ảnh môi trường cũng là một vấn đề cần quan tâm trong một thuật toán giấu tin. Rõ ràng là có thể chỉ giấu 1 bit thông tin vào mỗi ảnh mà không cần lo lắng về độ nhiễu của ảnh nhưng như vậy sẽ rất kém hiệu quả khi mà thông tin giấu có kích thước bằng Kb. Các thuật toán đều cố gắng đạt được mục đích làm thế nào giấu được nhiều thông tin nhất mà không gây ra nhiễu đáng kể.

1.1.7.3 Tính bảo mật

Thuật toán nhúng tin được coi là có tính bảo mật nếu thông tin được nhúng không bị tìm ra khi bị tấn công một cách có chủ đích trên cơ sở có hiểu biết đầy đủ về thuật toán nhúng tin và có bộ giải mã (trừ khóa bí mật), hơn nữa còn có được ảnh có mang thông tin (ảnh kết quả). Đây là một yêu cầu rất quan trọng đối với ảnh *image hiding.*

1.1.7.4 Ảnh môi trường đối với quá trình giải mã

Yêu cầu cuối cùng là thuật toán phải cho phép lấy lại được những thông tin đã giấu trong ảnh mà không có ảnh gốc. Điều này là một thuận lợi khi ảnh môi trường là duy nhất nhưng lại làm giới hạn khả năng ứng dụng của kỹ thuật giấu tin.

1.2. MỘT SỐ ẢNH ĐỊNH DẠNG BITMAP PHỔ BIẾN

1.2.1 Cấu trúc ảnh Bitmap

Ảnh BMP (*Bitmap*) được phát triển bởi *Microsoft Corporation*, được lưu trữ dưới dạng độc lập thiết bị cho phép *Windows* hiển thị dữ liệu không phụ thuộc vào khung chỉ định màu trên bất kì phần cứng nào. Tên *file* mở rộng mặc định của một *file* ảnh *Bitmap* là ".BMP". Ảnh BMP được sử dụng trên *Microsoft Windows* và các ứng dụng chạy trên *Windows* từ *version 3.0* trở lên.

Mỗi *file* ảnh *Bitmap* gồm 3 phần như bảng 1.1:

Bảng 1.1 *Cấu trúc ảnh BitMap*

Bitmap Header (54 byte)
Color Palette
Bitmap Data

1.2.1.1 Bitmap Header

Thành phần *bitcount* (Bảng 1.2) của cấu trúc *Bitmap Header* cho biết số bit dành cho mỗi điểm ảnh và số lượng màu lớn nhất của ảnh. *Bitcount* có thể nhận các giá trị sau:

- 1: *Bitmap* là ảnh đen trắng, mỗi bit biểu diễn 1 điểm ảnh. Nếu bit mang giá trị "0" thì điểm ảnh là điểm đen, nếu bit mang giá trị "1" thì điểm ảnh là điểm trắng.
- 4: *Bitmap* là ảnh 16 màu, mỗi điểm ảnh được biểu diễn bằng 4 bit.
- 8: *Bitmap* là ảnh 256 màu, mỗi điểm ảnh được biểu diễn bằng 8 bit.
- 16: *Bitmap* là ảnh *High Color*, mỗi dãy 2 byte liên tiếp trong *Bitmap* biểu diễn cường độ tương đối của màu đỏ, xanh lá cây và xanh lơ (RGB) của điểm ảnh.
- 24: *Bitmap* là ảnh *True Color*, mỗi dãy 3 byte liên tiếp trong *Bitmap* biểu diễn cường độ tương đối của màu đỏ, xanh lá cây và xanh lơ (RGB) của điểm ảnh.

Thành phần *Color Used* của cấu trúc *Bitmap Header* xác định số lượng màu của *Palete* thực sự được sử dụng để hiển thị Bitmap.Nếu thành phần này được đặt là 0, *Bitmap* sử dụng số màu lớn nhất tương ứng với giá trị của *bitcount*.

Bảng 1.2 *Thông tin về Bitmap Header*

Byte thứ	**Ý nghĩa**	**Giá trị**
1-2	Nhận dạng file	'BM' hay 19778
3-6	Kích thước file	Kiểu long trong Turbo C
7-10	Dự trữ	Thường mang giá trị 0
11-14	Byte bắt đầu vùng dữ liệu	Offset của byte bắt đầu vùng dữ liệu
15-18	Số byte cho vùng thông tin	4 byte
19-22	Chiều rộng ảnh BMP	Tính bằng pixel
23-26	Chiều cao ảnh BMP	Tính bằng pixel

27-28	Số Planes màu	Cố định là 1
29-30	Số bit cho 1 pixel (bitcount)	Có thể là: 1,4,8,16,24 tùy theo loại ảnh
31-34	Kiểu nén dữ liệu	0: Không nén 1: Nén runlength 8bits/pixel 2: Nén runlength 4bits/pixel
35-38	Kích thước ảnh	Tính bằng byte
39-42	Độ phân giải ngang	Tính bằng pixel / metter
43-46	Độ phân giải dọc	Tính bằng pixel / metter
47-50	Số màu sử dụng trong ảnh	
51-54	Số màu được sử dụng khi hiển thị ảnh (Color Used)	

1.2.1.2 Palette màu

Bảng màu của ảnh. Chỉ những ảnh nhỏ hơn hoặc bằng 8 bit mới có bảng màu.

***Bảng 1.3** Bảng màu của ảnh BITMAP*

Địa chỉ (Offset)	**Tên**	**Ý nghĩa**
0	RgbBlue	Giá trị cho màu xanh blue
1	RgbGreen	Giá trị cho màu xanh Green
2	RgbRed	Giá trị cho màu đỏ
3	RgbReserved	Dự trữ

1.2.1.3 Bitmap data

Phần này nằm ngay sau phần *Palete* màu của ảnh BMP. Đây là phần chứa giá trị màu của điểm ảnh trong ảnh BMP. Các dòng ảnh được lưu từ dưới lên trên, các điểm ảnh được lưu trữ từ trái sang phải. Giá trị của mỗi điểm ảnh là một chỉ số trỏ tới phần tử màu tương ứng trong *Palete* màu.

1.2.2 Cấu trúc ảnh PNG

1.2.2.1 Lịch sử và phát triển

Động cơ thúc đẩy cho việc tạo ra định dạng PNG bắt đầu vào khoảng đầu năm 1995, sau khi *Unisys* công bố họ sẽ áp dụng bằng sáng chế vào thuật toán nén dữ liệu LZW- được sử dụng trong định dạng GIF. Thuật toán được bảo vệ bởi bằng công nhận độc quyền sáng tạo ở Mỹ và tất cả các nước trên thế giới. Tuy nhiên, cũng đã có một số vấn đề với định dạng GIF khi cần có một số thay đổi nhất định trên hình ảnh, giới hạn của nó là 256 màu trong thời điểm máy tính có khả năng hiển thị nhiều hơn 256 màu đang trở nên phổ biến. Mặc dù định dạng GIF có thể thể hiện các hình ảnh động, song PNG vẫn được quyết định là định dạng hình ảnh đơn (chỉ có một hình duy nhất). Một người "anh em" của nó là MNG đã được tạo ra để giải quyết vấn đề ảnh động. PNG lại tăng thêm sự phổ biến của nó vào tháng 8 năm 1999, sau khi hãng *Unisys* huỷ bỏ giấy phép của họ đối với các lập trình viên phần mềm miễn phí, và phi thương mại.

- Phiên bản 1.0 của đặc tả PNG được phát hành vào ngày 1 tháng 7 năm 1996, và sau đó xuất hiện với tư cách RFC 2083. Nó được tổ chức W3C khuyến nghị vào ngày 1 tháng 10 năm 1996.
- Phiên bản 1.1, với một số thay đổi nhỏ và thêm vào 3 thành phần mới, được phát hành vào ngày 31 tháng 12 năm 1998.
- Phiên bản 1.2, thêm vào một thành phần mở rộng, được phát hành vào ngày 11 tháng 8 năm 1999.
- PNG giờ đây là một chuẩn quốc tế (ISO/IEC 15948:2003), và cũng được công bố như một khuyến nghị của W3C vào ngày 10 tháng 11 năm 2003. Phiên bản hiện tại của PNG chỉ khác chút ít so với phiên bản 1.2 và không có thêm thành phần mới nào.

1.2.2.2 Thông tin kỹ thuật

a. Phần đầu của tập tin

Một tập tin PNG bao gồm 8-byte kí hiệu (89 50 4E 47 0D 0A 1A) được viết trong hệ thống có cơ số 16, chứa các chữ "PNG" và hai dấu xuống dòng, ở giữa là sắp xếp theo số lượng của các thành phần, mỗi thành phần đều chứa thông tin về hình ảnh. Cấu trúc dựa trên các thành phần được thiết kế cho phép định dạng PNG có thể tương thích với các phiên bản cũ khi sử dụng.

b. Các "thành phần" trong tập tin

PNG là cấu trúc như một chuỗi các thành phần, mỗi thành phần chứa kích thước, kiểu, dữ liệu, và mã sửa lỗi CRC ngay trong nó.

Chuỗi được gán tên bằng 4 chữ cái phân biệt chữ hoa chữ thường. Sự phân biệt này giúp bộ giải mã phát hiện bản chất của chuỗi khi nó không nhận dạng được.

Với chữ cái đầu, viết hoa thể hiện chuỗi này là thiết yếu, nếu không thì ít cần thiết hơn (*ancillary*).Chuỗi thiết yếu chứa thông tin cần thiết để đọc được tệp và nếu bộ giải mã không nhận dạng được chuỗi thiết yếu, việc đọc tệp phải được hủy.

c. Thành phần cơ bản

Một bộ giải mã (*decoder*) phải có thể thông dịch để đọc và hiển thị một tệp PNG.

- IHDR phải là thành phần đầu tiên, nó chứa đựng *header*
- PLTE chứa đựng bảng màu (danh sách các màu)
- IDAT chứa đựng ảnh. Ảnh này có thể được chia nhỏ chứa trong nhiều phần IDAT. Điều này làm tăng kích cỡ của tệp lên một ít nhưng nó làm cho việc phát sinh ảnh PNG mượt hơn (*streaming manner*).
- IEND đánh dấu điểm kết thúc của ảnh.

1.3. PHƯƠNG PHÁP ĐÁNH GIÁ CHẤT LƯỢNG ẢNH SAU KHI GIẤU TIN

Để đánh giá chất lượng của bức ảnh (hay khung ảnh *video*) ở đầu ra của bộ mã hoá, người ta thường sử dụng hai tham số: Sai số bình phương trung bình – MSE (*Mean Square Error*) và phương pháp đề xuất với hệ số tỷ lệ tín hiệu / tín hiệu nhiễu PSNR (*Peak Signal to Noise Ratio*).

MSE giữa ảnh gốc và ảnh khôi phục được tính như sau:

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - y_{ij})^2$$

Ở đây:

x_{ij} biểu thị giá trị điểm ảnh gốc

y_{ij} biểu thị giá trị điểm ảnh đã được biến đổi

m và n lần lượt là chiều rộng và chiều cao của ảnh.

PSNR, đơn vị: *deciben* (dB), thường được sử dụng trong nghiên cứu xử lý hình ảnh:

$$\text{PSNR} = 10*\log10(\frac{255^2}{MSE})$$

Thông thường, nếu PSNR > 35dB thì hệ thống mắt người gần như không phân biệt được giữa ảnh gốc và ảnh khôi phục. PSNR càng cao thì chất lượng ảnh khôi phục càng tốt. Khi hai hình ảnh giống hệt nhau, MSE sẽ bằng 0 và PSNR đi đến vô hạn.

Chương 2. KỸ THUẬT GIẤU TIN THUẬN NGHỊCH TRONG ẢNH

2.1. KHÁI NIỆM GIẤU TIN THUẬN NGHỊCH

2.1.1. Khái niệm

Theo đặc tính tách tin cần lưu trữ hay không lưu trữ vật mang tin mà người ta phân kỹ thuật giấu tin ra làm 2 loại:

- Kỹ thuật giấu không thuận nghịch: Là kỹ thuật giấu sau khi tách thông điệp không thể khôi phục lại ảnh gốc. Những kỹ thuật này phục vụ trao đổi thông tin mật. Người ta có thể hủy vật mang tin khi cần thiết mà không cần lưu trữ.
- Kỹ thuật giấu thuận nghịch: Là kỹ thuật giấu sau khi tách thông điệp có thể khôi phục lại ảnh gốc. Những kỹ thuật này phục vụ trong một số lĩnh vực như: y học, quân sự, nghiên cứu năng lượng hoặc hệ thống thông tin vệ tinh, …

2.1.2. Một số kỹ thuật giấu thuận nghịch điển hình

Năm 1999, *Honsinger* và các cộng sự đề xuất kỹ thuật giấu thuận nghịch đầu tiên [2], mở ra một hướng mới trong lĩnh vực giấu tin. Tiếp đó một loạt các kỹ thuật giấu tin thuận nghịch khác được công bố. Sau đây giới thiệu sơ lược một số kỹ thuật giấu tiêu biểu.

Kỹ thuật mở rộng sai phân DE (*Difference Expansion*) do *Tian* đưa ra [3], đây là kỹ giấu tin dựa trên mở rộng hệ số sai phân của điểm ảnh, dữ liệu ảnh được tính sai phân theo biểu thức $D_i=I_i - I_{i+1}$ (I_i là giá trị *pixel* của ảnh), thông tin được giấu trên LSB của các hệ số sai phân sau khi được mở rộng. Sau đó tác giả đề xuất tiếp phương pháp mở rộng trên các hệ số *wavelet* để giấu tin [4]. Đến năm 2008, *Shaowei Weng* và các đồng nghiệp đưa ra kỹ thuật DE cải tiến [5] bằng cách thêm vào hàm nén – giãn trong quá trình giấu tin sử dụng DE nhằm giảm nhiễu xẩy ra (theo đánh giá bằng PSNR) của kỹ thuật giấu thuận nghịch DE.

Năm 2003, Ni và cộng sự đề xuất kỹ thuật giấu thuận nghịch dựa trên dịch chuyển biểu đồ tần suất gọi là NSAS [6]. Tiếp đó một loạt các kỹ thuật giấu thuận nghịch dựa phương pháp này ra đời: kỹ thuật DIH [7] (dịch chuyển biểu đồ tần suất hệ số sai phân), kỹ thuật HKC [8] (cải tiến kỹ thuật giấu NSAS).

2.1.1.1 Phương pháp giấu NSAS.

Thuật toán nhúng.

Đầu vào: Một ảnh cấp xám C, một chuỗi thông điệp M.

Đầu ra: Ảnh giấu tin S.

Bước 1: Quét tất cả các ảnh và xây dựng biểu đồ tần số $H_1(x)$, $x \in |0, 255|$. Trong histogram, giá trị màu xám cao nhất được kí hiệu là a, giá trị màu xám thấp nhất được kí hiệu là b. $H_1(b) = 0$, thì b được gọi là một điểm cực tiểu. Để đơn giản, giả sử a < b.

Bước 2: Quét ảnh 1 lần nữa và ghi lại các giá trị điểm ảnh = b và đặt chúng vào bản đồ L. Sau đó, thay đổi biểu đồ tần số $H_1(x)$, $x \in |0, 255|$ sang bên phải một đơn vị để trống cột tần suất tại vị trí có giá trị a+1.

Bước 3: Trích một bit dữ liệu từ dữ liệu bí mật S. Quét tất cả các ảnh 1 lần nữa. Nếu quét giá trị các điểm ảnh và các bit dữ liệu nhúng là 1, thì đặt giá trị điểm ảnh là a+1. Nếu bit dữ liệu được nhúng là 0, thì không thay đổi các điểm ảnh được quét.

Bước 4: lặp lại bước 3 cho đến khi dữ liệu S được nhúng hoàn toàn.

Thuật toán tách.

Đầu vào: Ảnh Giấu tin S.

Đầu ra: Ảnh khôi khục và chuỗi thông điệp M.

Bước 1: Quét tất cả các ảnh theo thứ tự như trong giai đoạn nhúng. Nếu quét được giá trị a, thì tách bit 0 khỏi a. Nếu quét được giá trị a+1 thì tách bit 1 ra khỏi a.

Bước 2: Quét tất cả cá ảnh 1 lần nữa và dịch chuyển $H_1(x)$, $x \in |0, 255|$ sang trái 1 đơn vị.

Bước 3: thiết lập các giá trị các giá trị ghi được trong bản đồ L là b.

2.1.1.2. Thuật toán cải tiến NSAS.

Thuật toán nhúng tin.

Đầu vào: Một ảnh cấp xám C, chuỗi thông điệp M.

Đầu ra: Ảnh giấu tin S.

Bước 1: Quét tất cả các ảnh và xây dựng biểu đồ tần suất $H_1(x)$, $x \in |0, 255|$. Trong histogram, có điểm cực đại a, điểm cực tiểu b. Không mất tính khái quát, giả sử a < b.

Bước 2: Thiết lập k = 0. Giá trị k được sử dụng để cho biết số bít dữ liệu nhúng.

Bước 3: Quét tất cả các ảnh 1 lần nữa. Nếu quét được giá trị điểm ảnh = 1, trích 1 bit dữ liệu từ S, thiết lập k = k +1 và tiếp tục bước 4 để nhúng dữ liệu S, nếu không, thực hiện bước 5.

Bước 4: Nếu bit dữ liệu là 1, thì thiết lập giá trị điểm ảnh quét được là a+1, nếu không có thay đổi gì cho những điểm ảnh này, quay lại bước 3 tiếp tục quá trình nhúng.

Bước 5: Nếu tất cả các giá trị điểm ảnh quét được nằm trong khoảng (a, b), thì cộng các giá trị điểm ảnh đó thêm 1. Ghi lại vị trí các điểm ảnh có giá trị điểm ảnh bằng b.

Thuật toán tách tin.

Đầu vào: Ảnh giấu tin S.

Đầu ra: Ảnh khôi phục và chuỗi thông điệp M.

Bước 1: Thiết lập k = 0.

Bước 2: Quét tất cả các ảnh theo thứ tự như trong quá trình nhúng. Nếu quét được giá trị là a, thì đặt k = k+1 và tách bit 0 khỏi a. Nếu quét được giá trị là a+1, thì k = a+1 và tách bit 1 ra khỏi a. Nếu giá trị quét nằm trong khoảng (a, b) thì các giá trị điểm ảnh quét được trừ đi 1. Nếu vị trí các điểm ảnh được ghi trong bản đồ L, thì thiết lập giá trị các điểm ảnh quét được là b.

Bước 3: Lặp lại bước 2 cho đến khi k = |S|

So sánh phương pháp NSAS và NSAS cải tiến để thấy được sự hiệu quả của phương pháp cái tiến NSAS. Một ảnh màu xám 8 bit: *Baboon* kích thước 512x512 được lựa chọn để thí nghiệm, dùng PSNR để đánh giá.

2.2. KỸ THUẬT GIẤU THUẬN NGHỊCH BẰNG THUẬT TOÁN MAXMIN

2.2.1. Giới thiệu

Thuật toán **MAXMIN** do *Chingyu YANG* đề xuất năm 2012 [1]. Thuật toán **MAXMIN** bao gồm hai phần: Thuật toán giấu tin bằng bảo toàn nhỏ nhất và thuật toán giấu tin bằng bảo toàn lớn nhất. Phương pháp thực hiện bằng cách chia ảnh thành nhiều khối nhỏ, trong mỗi khối tính sai phân dựa vào giá trị lớn nhất, nhỏ

nhất của khối, sau đó để tránh vượt ngưỡng tăng giảm giá trị của sai phân dựa vào một giá trị điều khiển cho trước rồi chèn thông điệp vào các giá trị sai phân này.

Thông thường thuật toán giấu tin bằng bảo toàn nhỏ nhất thì ít hao tổn hơn khi nhúng một thông tin bí mật vào một loạt các hình ảnh. Thuật toán bảo toàn lớn nhất thay thế các thuật toán giấu tin bằng bảo toàn nhỏ nhất khi thuật toán giấu tin bằng bảo toàn nhỏ nhất không có khả năng tiến hành khôi phục lại dữ liệu ẩn trên một hình ảnh nhất định. Để cung cấp một sự lưu trữ ẩn cao hơn và khắc phục các vấn đề vượt ngưỡng, thuật toán **MAXMIN** đã nhúng các bit dữ liệu vào một khối khác biệt mà chúng được tạo ra bằng cách trừ tối thiểu (hoặc tối đa) các giá trị *pixel* từ các điểm ảnh còn lại của khối. Các ví dụ cho thấy rằng phương pháp này không chỉ hoàn toàn phục hồi môi trường giấu tin mà còn tạo ra một chất lượng nhận diện cao của các hình ảnh được đánh dấu. Hơn nữa, hiệu suất tải trọng và PSNR của phương pháp này vượt trội hơn hẳn so với các chương trình hiện có. Hơn nữa, phương pháp này có khả năng xử lý các loại hình ảnh khác nhau mà không có bất kỳ việc xảy ra tràn ngưỡng, ngược lại một số trong số những hình ảnh đó còn có thể gây khó khăn cho nhiều kỹ thuật hiện có khác.

Mục 2.2.2 trình bày chi tiết các bước của thuật toán giấu tin thuận nghịch dựa trên bảo toàn lớn nhất, nhỏ nhất.

2.2.2. Thuật toán

2.2.2.1. Thuật toán giấu tin bằng bảo toàn nhỏ nhất

Quá trình giấu tin:

- Đầu vào :
  - Ảnh sử dụng để giấu tin C
  - Thông tin cần giấu M
- Đầu ra :
  - Ảnh đã giấu tin S
- Các bước thực hiện :

- Bước 1: Tách ảnh đầu vào C thành các khối p_j=$\{p_{ij}\}_{i=0}^{n\times n-1}$ kích cỡ n×n.
- Bước 2: Tính sai phân của từng khối theo công thức :

$$\hat{p}_{ij}{}_{i=0}^{n\times n-1} = \{p_{ij}\}_{i=0}^{n\times n-1} - \theta_j$$

Trong đó θ_j =Minarg $\{p_{ij}\}_{i=0}^{n\times n-1}$ là giá trị nhỏ nhất của khối p_j.

Thông điệp cần giấu M được chuyển sang chuỗi nhị phân $\{b_k\}_{k=0}^{L}$

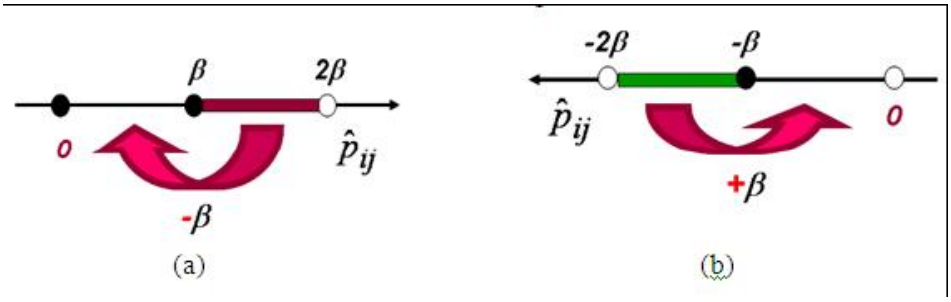
- Bước 3: Thực hiện phân chia các giá trị sai phân của khối để được $\tilde{p}_{ij}$ theo biểu thức điều kiện sau : $\tilde{p}_{ij} = \hat{p}_{ij} - \beta$ nếu $\beta \leq \tilde{p}_{ij} < 2\beta$

  Trong đó β là tham số điều khiển.(Theo minh họa hình 2.1(a))

  Tạo 1 bản đồ cờ đánh giấu các vị trí sai phân đã dịch chuyển để có thể khôi phục ảnh gốc trong quá trình tách tin.

  Bước 4: Thực hiện giấu tin vào các $\tilde{p}_{ij}$ theo biểu thức $p'_{ij} = \bar{p}_{ij} \times 2$ ($\bar{p}_{ij} \in \{\hat{p}_{ij}, \tilde{p}_{ij}\}$) và chèn bit cần giấu vào LSB của $\bar{p}_{ij}$. (Hoặc $\bar{p}_{ij} + b_k$ với b_k là bit thông điệp cần giấu)

- Bước 5: Khôi phục lại giá trị *pixel* của ảnh đã giấu tin được S dựa vào θ_j và hệ số sai phân của khối.



Hình 2.1.Cấu trúc của việc điều chỉnh điểm ảnh (a) Chương trình bảo vệ tối thiểu, (b) chương trình bảo vệ tối đa.

Quá trình tách tin:

- Đầu vào :
  - Ảnh đã giấu tin S
- Đầu ra :
  - Thông tin đã giấu M
  - Ảnh gốc C
- Các bước thực hiện :

- Bước 1: Tách ma trận điểm ảnh của S thành các khối $Q_j = \{q_{ij}\}_{i=0}^{n\times n-1}$ kích cỡ n×n.
- Bước 2: Tính sai phân của từng khối theo công thức :

$$\hat{q}_{ij}{}_{i=0}^{n\times n-1} = \{q_{ij}\}_{i=0}^{n\times n-1} - \theta_j$$

  Trong đó θ_j =Minarg $\{q_{ij}\}_{i=0}^{n\times n-1}$ là giá trị nhỏ nhất của khối p_j.

- Bước 3: Bit thông tin đã giấu được tách ra từ các giá trị sai phân thỏa mãn điều kiện $0 \leq \hat{q}_{ij} < 2\beta$ bằng phép toán chia lấy dư của $\hat{q}_{ij}$ cho 2.
- Bước 4: Giá trị pixel của ảnh gốc ban đầu được khôi phục bằng công thức:

$$\hat{p}_{ij} = \left\lfloor \frac{\hat{q}_{ij}}{2} \right\rfloor$$

Trong đó $\lfloor x \rfloor$ là hàm làm tròn về -∞.

Tiếp theo $\hat{q}_{ij}$ được cộng thêm β nếu cờ tương ứng trong khối bằng 1.

- Bước 5: Lặp lại các bước trên cho đến khi tách hết các bit đã giấu M

2.2.2.2. Thuật toán giấu tin bằng bảo toàn lớn nhất

Quá trình giấu tin:

- Đầu vào :
  - Ảnh sử dụng để giấu tin C
  - Thông tin cần giấu M
- Đầu ra :
  - Ảnh đã giấu tin S
- Các bước thực hiện :

- Bước 1: Tách ảnh đầu vào C thành các khối p_j=$\{p_{ij}\}_{i=0}^{n\times n-1}$ kích cỡ n×n.
- Bước 2: Tính sai phân của từng khối theo công thức :

$$\hat{p}_{ij}{}_{i=0}^{n\times n-1} = \{p_{ij}\}_{i=0}^{n\times n-1} - \theta_j$$

Trong đó θ_j =Minarg $\{p_{ij}\}_{i=0}^{n\times n-1}$ là giá trị lớn nhất của khối p_j.

Thông điệp cần giấu M được chuyển sang chuỗi nhị phân $\{b_k\}_{k=0}^{L}$

- Bước 3: Thực hiện phân chia các giá trị sai phân của khối để được $\tilde{p}_{ij}$ theo biểu thức điều kiện sau : $\tilde{p}_{ij} = \hat{p}_{ij} + \beta$ nếu $-2\beta < \tilde{p}_{ij} \leq -\beta$
Trong đó β là tham số điều khiển.(Theo minh họa hình 2.1 (b))

Tạo 1 bản đồ cờ đánh giấu các vị trí sai phân đã dịch chuyển để có thể khôi phục ảnh gốc trong quá trình tách tin.

- Bước 4: Thực hiện giấu tin vào các $\tilde{p}_{ij}$ theo biểu thức p'_{ij}= $\bar{p}_{ij} \times 2$ ($\bar{p}_{ij} \in \{\hat{p}_{ij}, \tilde{p}_{ij}\}$) và chèn bit cần giấu vào LSB của $\bar{p}_{ij}$. (Hoặc $\bar{p}_{ij} + b_k$ với b_k là bit thông điệp cần giấu)
- Bước 5: Khôi phục lại giá trị *pixel* của ảnh đã giấu tin được ảnh S dựa vào θ_j và hệ số sai phân của khối.

Quá trình tách tin:

- Đầu vào :
  - Ảnh đã giấu tin S

- Đầu ra :
  - Thông tin đã giấu M
  - Ảnh gốc C

- Các bước thực hiện :

- Bước 1: Tách ma trận điểm ảnh của S thành các khối Q_j=$\{q_{ij}\}_{i=0}^{n\times n-1}$ kích cỡ n×n.
- Bước 2: Tính sai phân của từng khối theo công thức :

$$\hat{q}_{ij}{}_{i=0}^{n\times n-1} = \{q_{ij}\}_{i=0}^{n\times n-1} - \theta_j$$

Trong đó θ_j =Minarg $\{q_{ij}\}_{i=0}^{n\times n-1}$ là giá trị lớn nhất của khối p_j.

- Bước 3: Bit thông tin đã giấu được tách ra từ các giá trị sai phân thỏa mãn điều kiện -2β< $\hat{q}_{ij} \leq 0$ bằng phép toán chia lấy dư của $\hat{q}_{ij}$ cho 2.
- Bước 4: Giá trị *pixel* của ảnh gốc ban đầu C được khôi phục bằng công thức

$$\hat{p}_{ij} = \left\lfloor \frac{\hat{q}_{ij}}{2} \right\rfloor$$

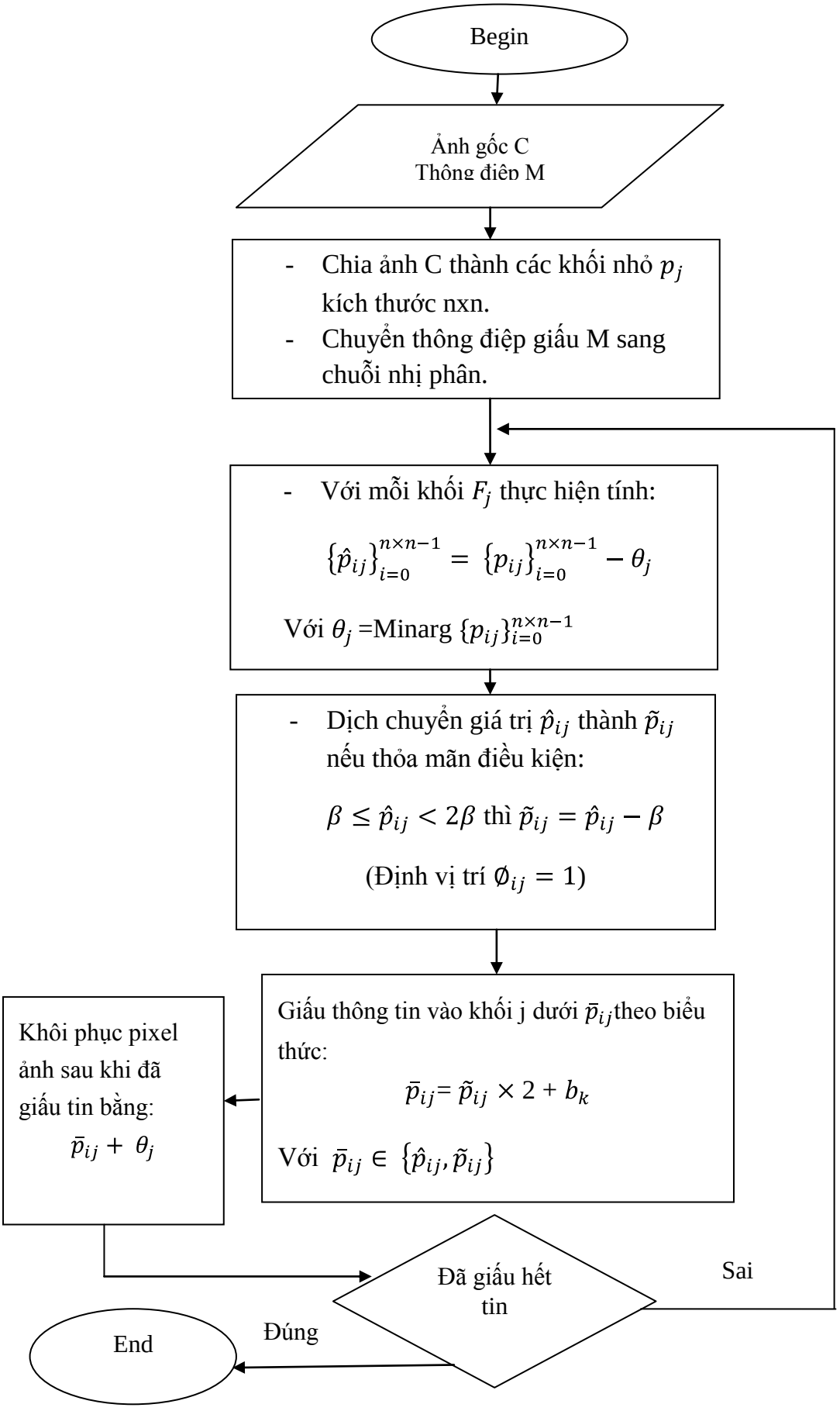
Trong đó $\lfloor x \rfloor$ là hàm làm tròn về -∞.

Tiếp theo $\hat{q}_{ij}$ được trừ cho β nếu cờ tương ứng trong khối bằng 1.

- Bước 5: Lặp lại các bước trên cho đến khi tách hết các bit đã giấu M.

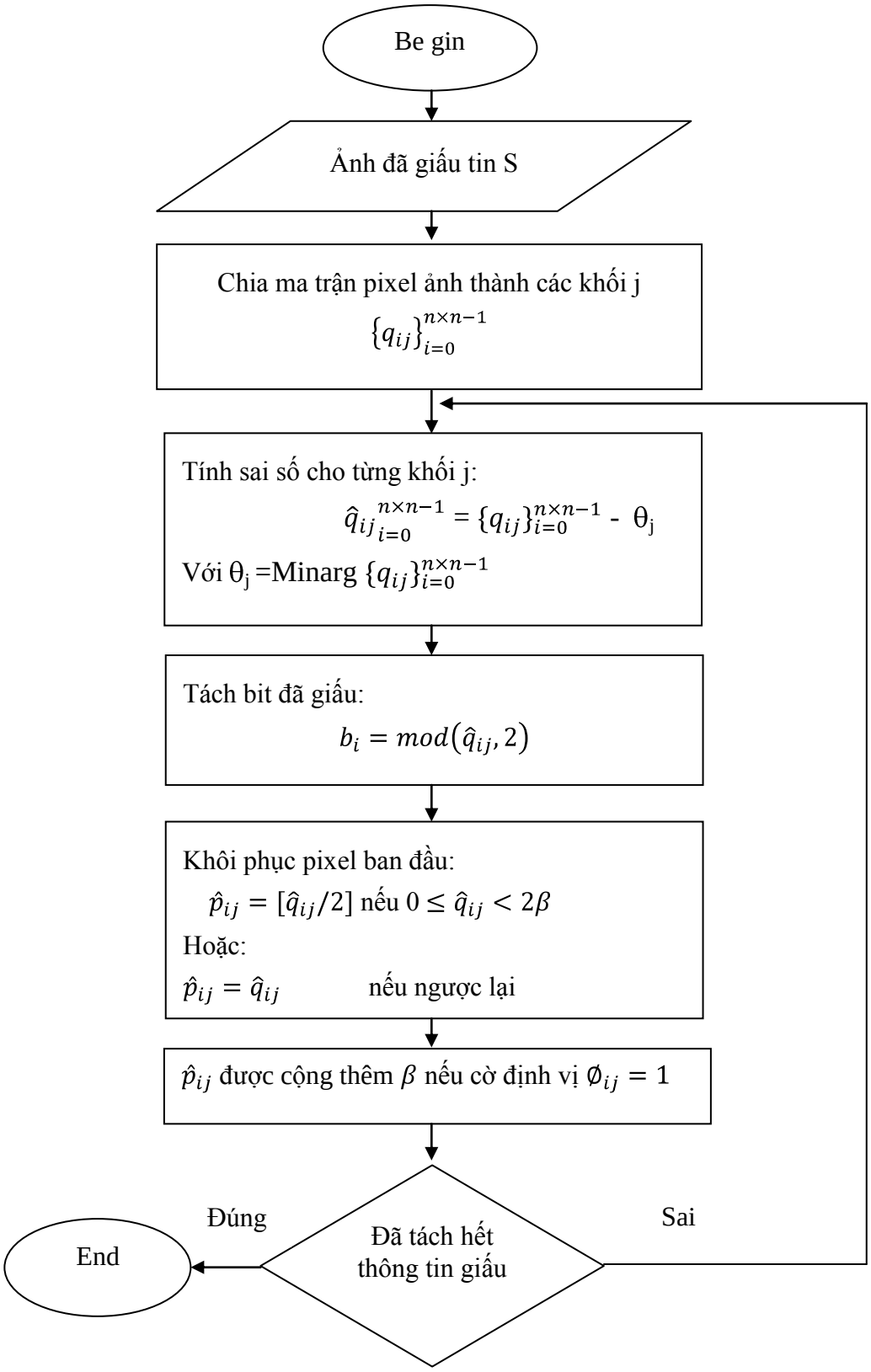
2.2.3. Lược đồ giấu tin và tách tin.

2.2.3.1. *Lược đồ giấu tin bảo toàn nhỏ nhất.*



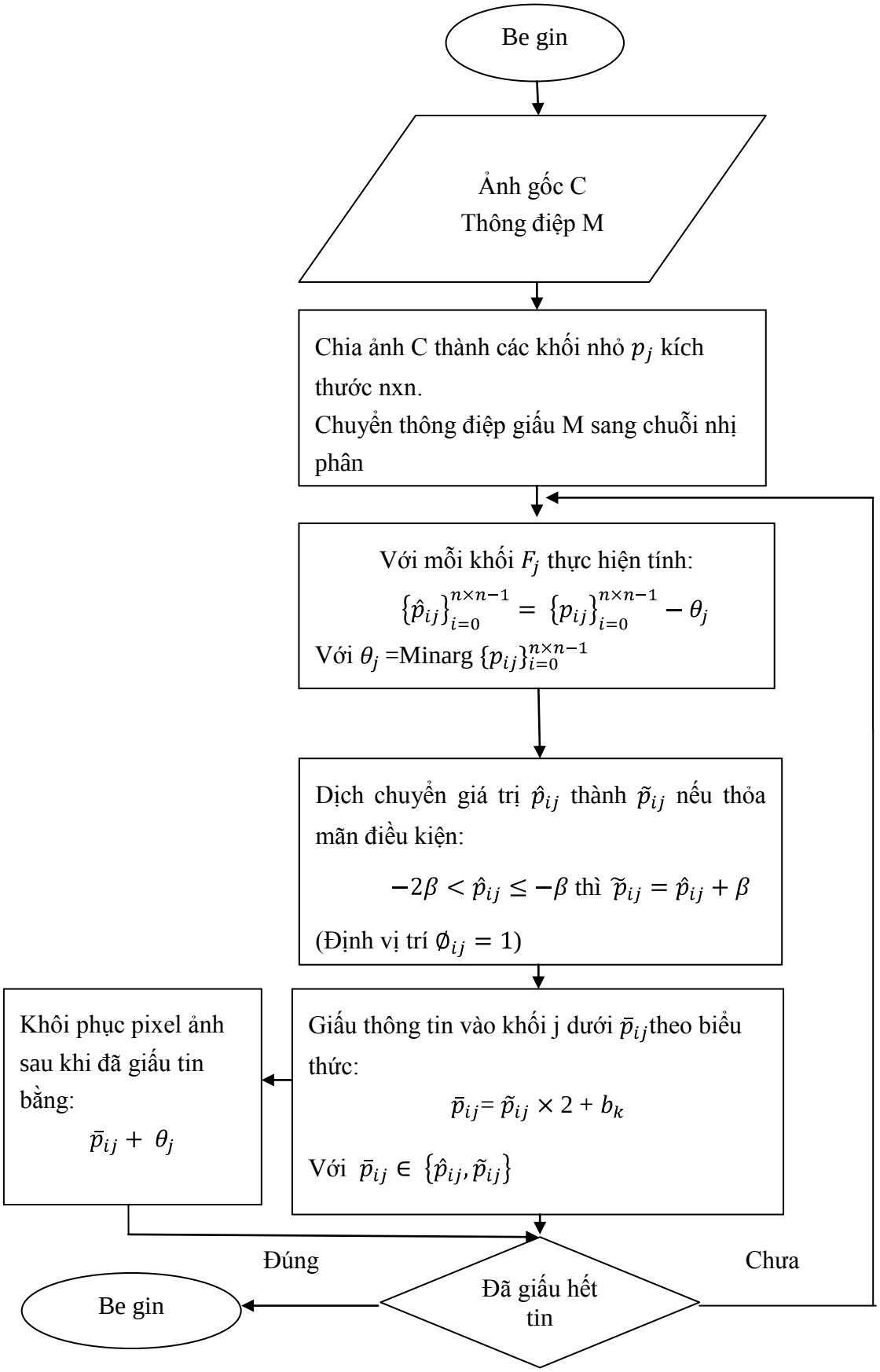
Hình 2.2. Lược đồ giấu tin bảo toàn nhỏ nhất

2.2.3.2. Lược đồ tách tin bảo toàn nhỏ nhất.



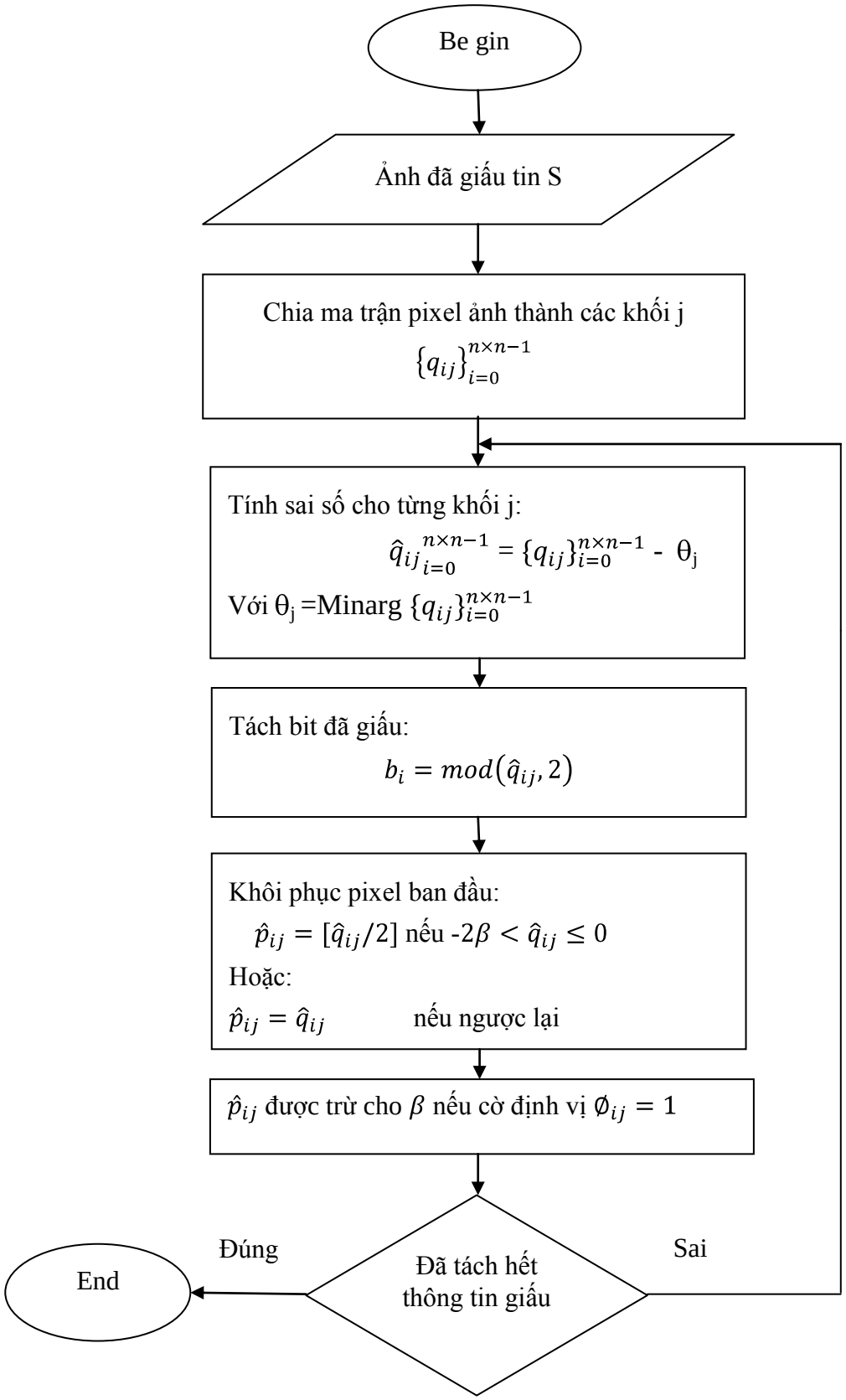
Hình 2.3. Lược đồ tách tin bảo toàn nhỏ nhất

2.2.3.3. ***Lược đồ giấu tin bảo toàn lớn nhất.***



Hình 2.4. Lược đồ giấu tin bằng bảo toàn lớn nhất

2.2.3.4. Lược đồ tách tin bảo toàn lớn nhất.



Hình 2.5. Lược đồ tách tin bảo toàn lớn nhất

2.2.4. Ví dụ minh họa.

- Đầu vào:
  - Ảnh có kích cỡ 6×6 :

167	167	168	153	154	158
165	165	167	153	152	154
166	163	168	154	155	156
167	163	168	166	166	160
165	165	166	165	160	162
165	163	168	161	163	159

  - Thông tin cần giấu là chuỗi M = 'BOM NO'
  - Tham số điều khiển: β =4
- Đầu ra:
  - Ảnh được giấu tin
- Các bước thực hiện:
  - **Chương trình bảo toàn nhỏ nhất:**

- Bước 1. Chia ảnh thành các khối nhỏ 3×3

167	167	168	153	154	158	167	163	168	166	166	160
165	165	167	153	152	154	165	165	166	165	160	162
166	163	168	154	155	156	165	163	168	161	163	159

- Bước 2. Tìm Min, tính sai phân

  Chuyển chuỗi văn bản sang nhị phân:

  'BOM NO'=010000100100111101001101001000000100111001001111

4	4	5	1	2	6	4	0	3	7	7	1
2	2	4	1	152	5	2	2	3	6	1	3
3	163	5	2	3	4	2	163	5	2	4	159

- Bước 3. Phân chia $\tilde{p}_{ij}$ các sai phân sẽ giấu dựa vào β

[0]	[0]	[1]	1	2	[2]	[0]	0	3	[3]	[3]	1
2	2	[0]	1	152	2	2	2	3	[2]	1	3
3	163	[1]	2	3	[0]	2	163	[1]	2	[4]	159

- Bước 4. Nhân $\tilde{p}_{ij}$ và $\hat{p}_{ij}$ với 2 và giấu tin bằng cách cộng thêm bit thông tin vào $\tilde{p}_{ij}$ ta được

1	0	2	2	4	5	0	1	7	6	6	2
4	4	1	3	152	5	4	5	6	4	3	6
6	163	3	5	7	0	5	163	2	4	9	159

- Bước 5. Khôi phục lại ảnh, được ảnh đã giấu tin

164	163	165	154	156	157	163	164	170	165	165	161
167	167	164	155	152	157	167	168	169	163	162	165
169	163	166	157	159	152	168	163	165	163	168	159

- **Chương trình bảo toàn lớn nhất.**

- Bước 1. Chia ảnh thành các khối nhỏ 3×3 ta được 4 khối.

167	167	168	153	154	158	167	163	168	166	166	160
165	165	167	153	152	154	165	165	166	165	160	162
166	163	168	154	155	156	165	163	168	161	163	159

- Bước 2. Tìm Max, tính sai phân

  Chuyển chuỗi văn bản sang nhị phân:
  'BOM NO'=010000100100111101001101001000000100111001001111

-1	-1	168	-5	-4	158	-1	-5	168	166	0	-6
-3	-3	-1	-5	-6	-4	-3	-3	-2	-1	-6	-4
-2	-5	0	-4	-3	-2	-3	-5	0	-5	-3	-7

- Bước 3. Phân chia $\tilde{p}_{ij}$ các sai phân sẽ giấu dựa vào β

-1	-1	168	–1	0	158	-1	–1	168	166	0	–2
-3	-3	-1	–1	–2	0	-3	-3	-2	-1	–2	0
-2	–1	0	0	-3	-2	-3	–1	0	–1	-3	–3

- Bước 4. Nhân $\tilde{p}_{ij}$ và $\hat{p}_{ij}$ với 2 và giấu thông tin bằng cách trừ $\tilde{p}_{ij}$ với bit thông tin

-3	-2	168	-2	0	158	-2	-3	168	166	0	-4
-6	-6	-2	-3	-5	-1	-7	-6	-5	-2	-4	-1
-5	-2	1	-1	-7	-4	-6	-3	0	-2	-6	-7

- Bước 5. Khôi phục lại ảnh, được ảnh đã giấu tin

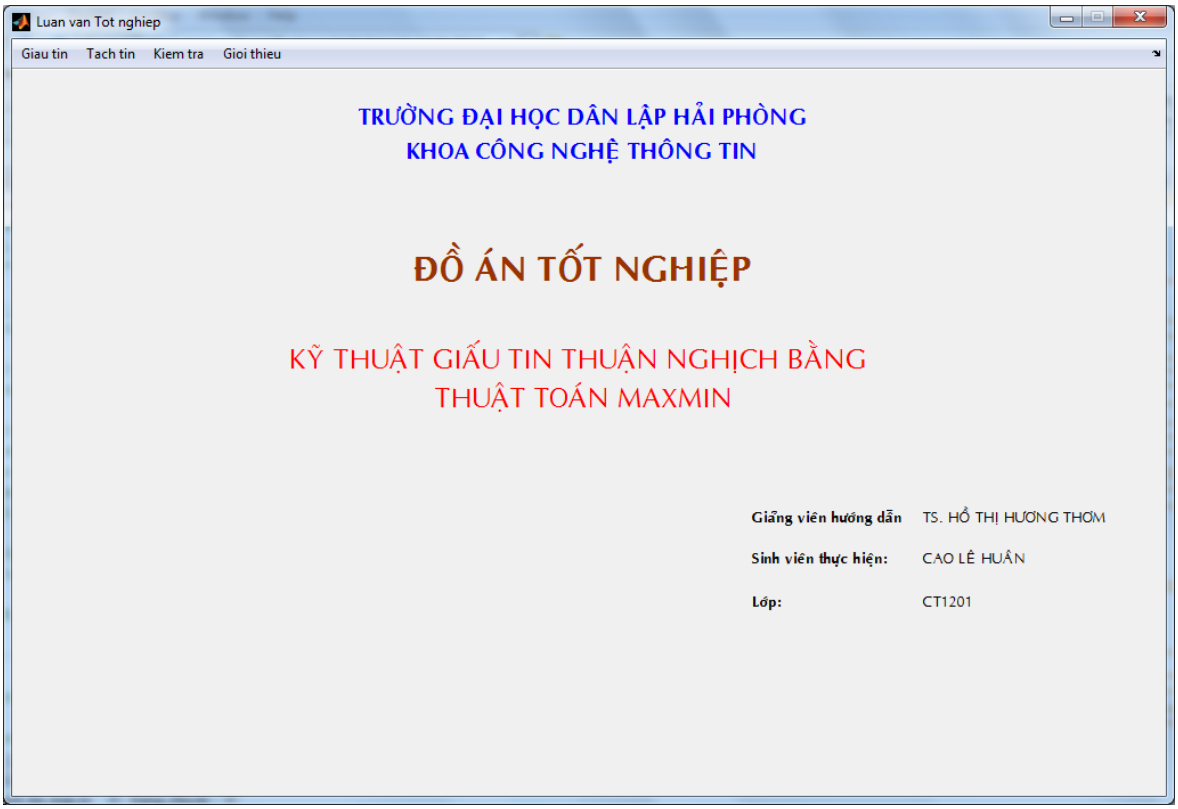
165	166	168	156	158	158	166	165	168	166	166	162
162	162	166	155	153	157	161	162	163	164	162	165
163	166	167	157	151	154	162	165	168	164	160	159

Chương 3. CÀI ĐẶT THỬ NGHIỆM

3.1. MÔI TRƯỜNG CÀI ĐẶT.

- Ngôn ngữ cài đặt, môi trường soạn thảo và chạy chương trình được thực hiện trên ngôn ngữ lập trình Matlab 7.12.0.635(R2011a)
- Hệ điều hành Window 7

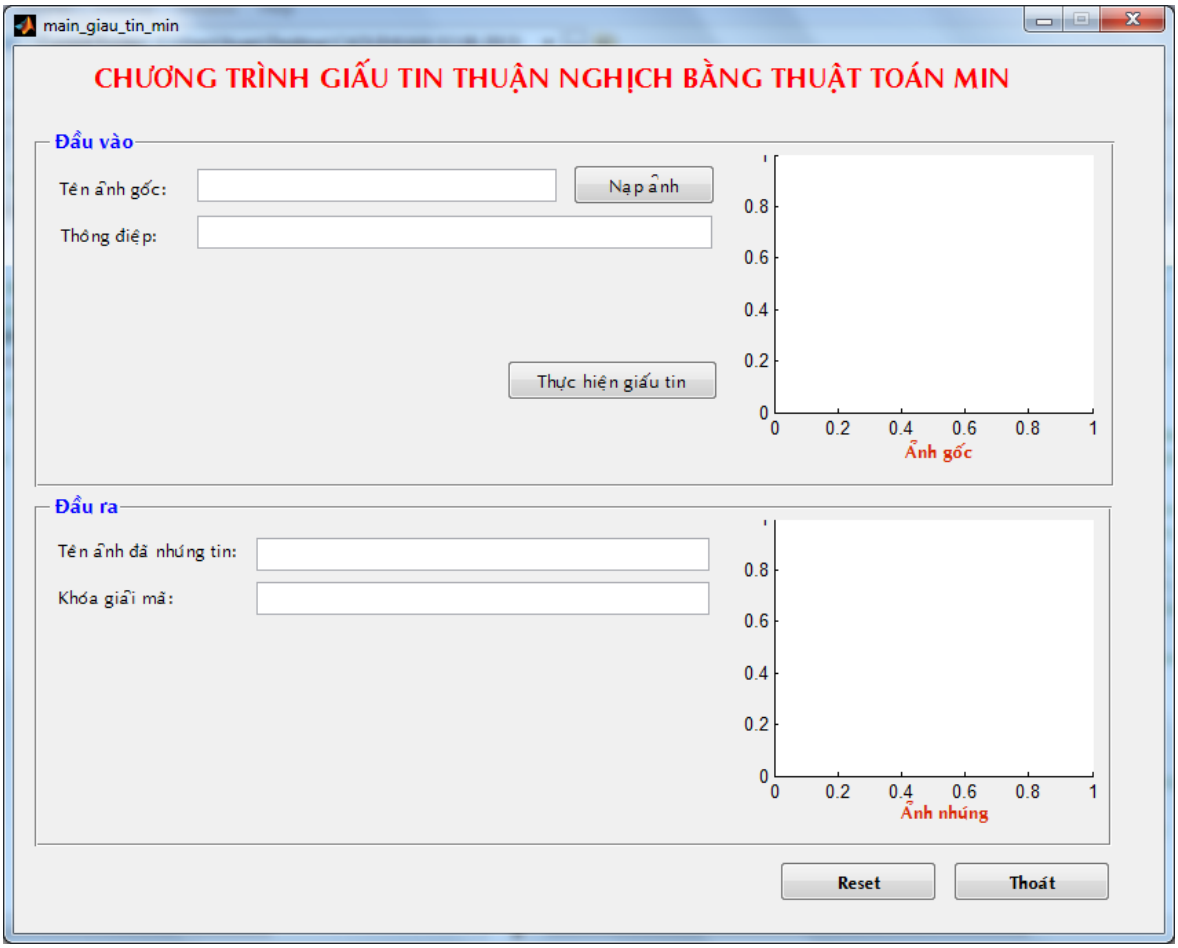
3.2. GIAO DIỆN CHƯƠNG TRÌNH.



Hình 3.1 Giao diện chính của chương trình

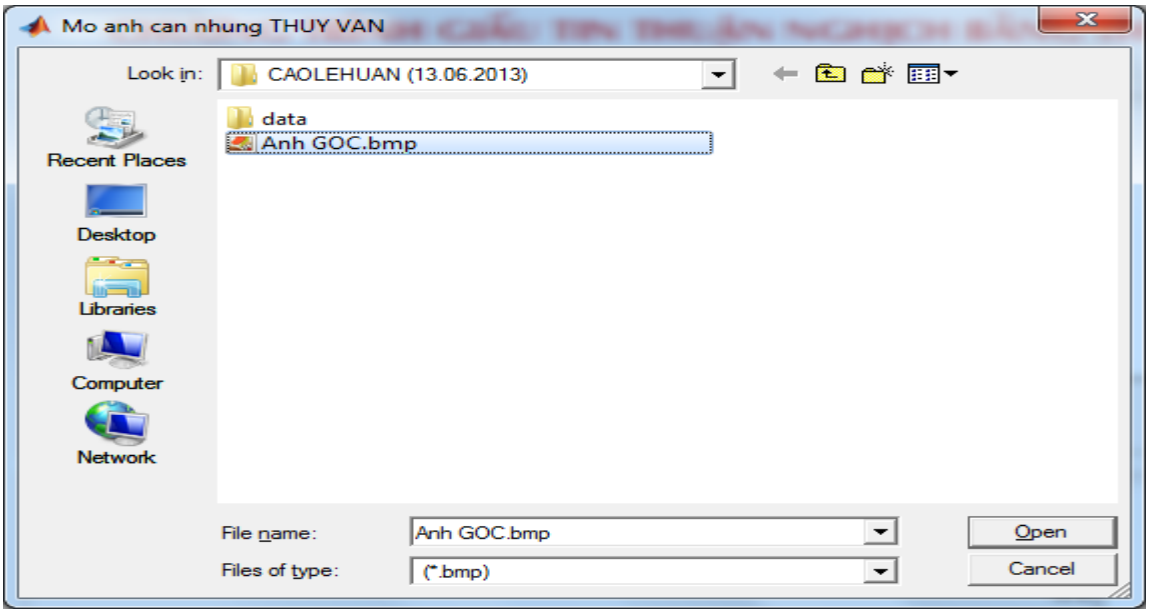
Đây là giao diện chính của chương trình, từ đây ta sẽ gọi đến các giao diện khác thông qua menu.

Từ menu "Giau tin" chọn "Thuat toan Min" sẽ gọi đến giao diện:



Hình 3.2 Giao diện giấu tin bằng thuật toán Min.

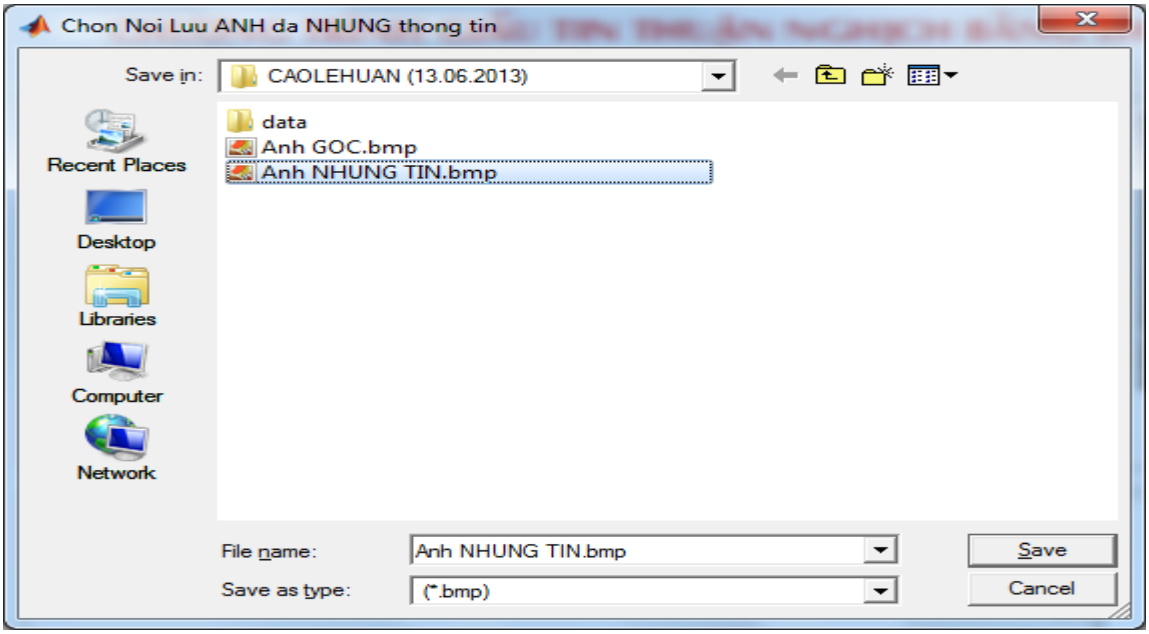
Đây là giao diện dùng để giấu tin bằng thuật toán Min. Để nhập ảnh ta kích vào nút "Nạp ảnh" một hộp thoại sẽ hiện ra để ta chọn ảnh cần giấu tin:



Hình 3.3 Chọn ảnh giấu tin

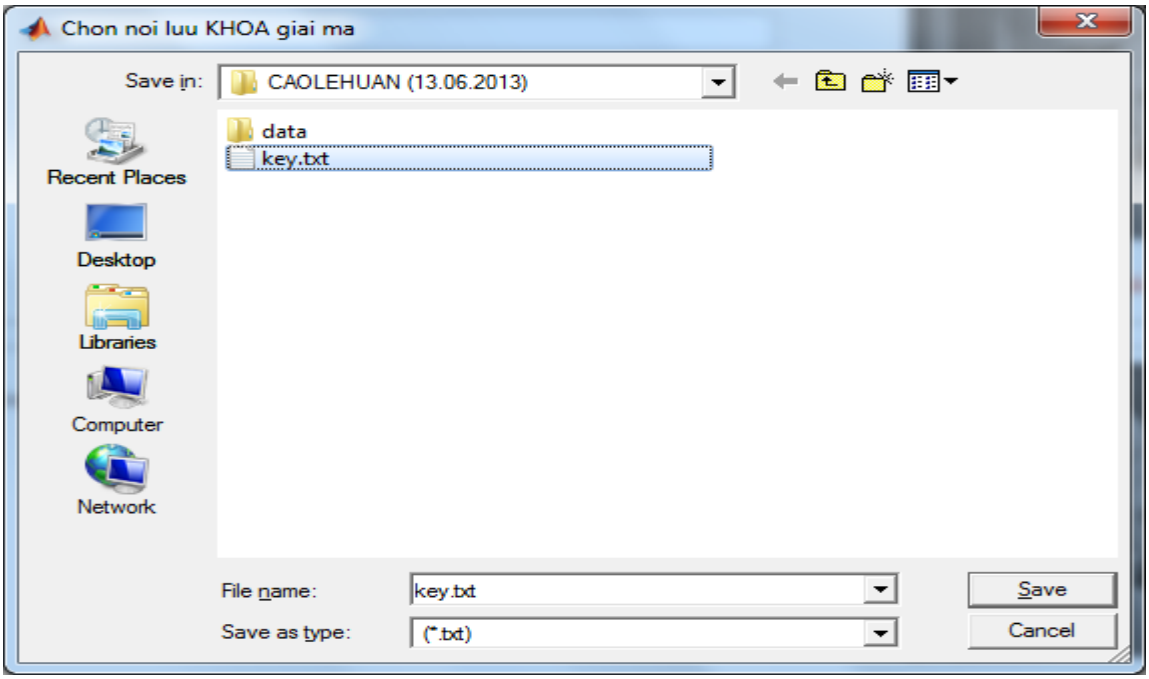
Sau khi chọn ảnh giấu tin sẽ có thông báo "da NAP xong ANH GOC". Tiếp theo ta nhập thông tin cần giấu vào ô "Thông điệp".

Đến phần giấu tin ta kích vào nút "Thực hiện giấu tin" sẽ có 1 hộp thoại xuất hiện để chọn nơi lưu ảnh đã nhúng tin:



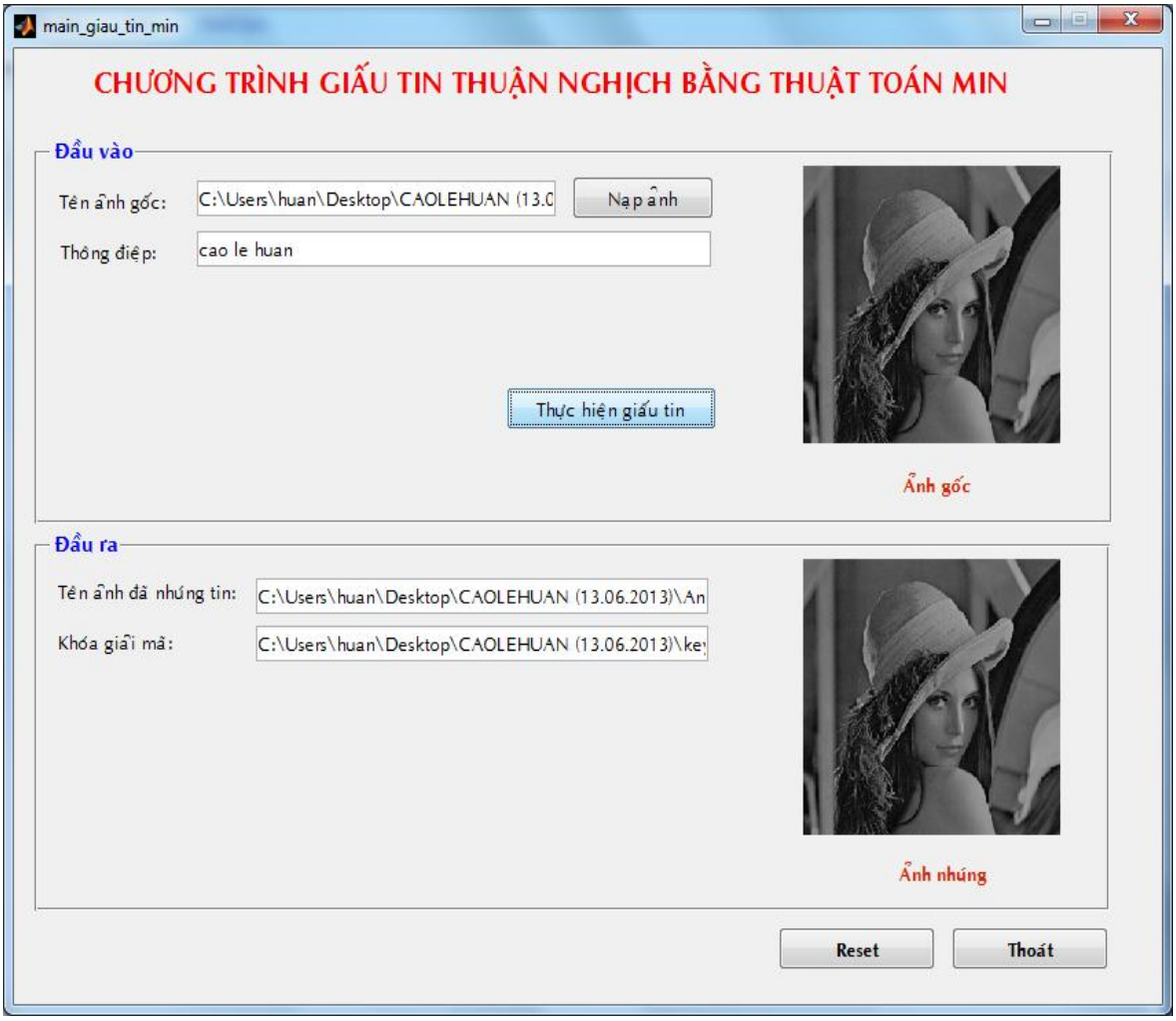
Hình 3.4 Nơi lưu ảnh đã nhúng thông tin

Sau khi lưu ảnh đã nhúng tin sẽ xuất hiện hộp thoại để chọn nơi lưu khóa giải mã:



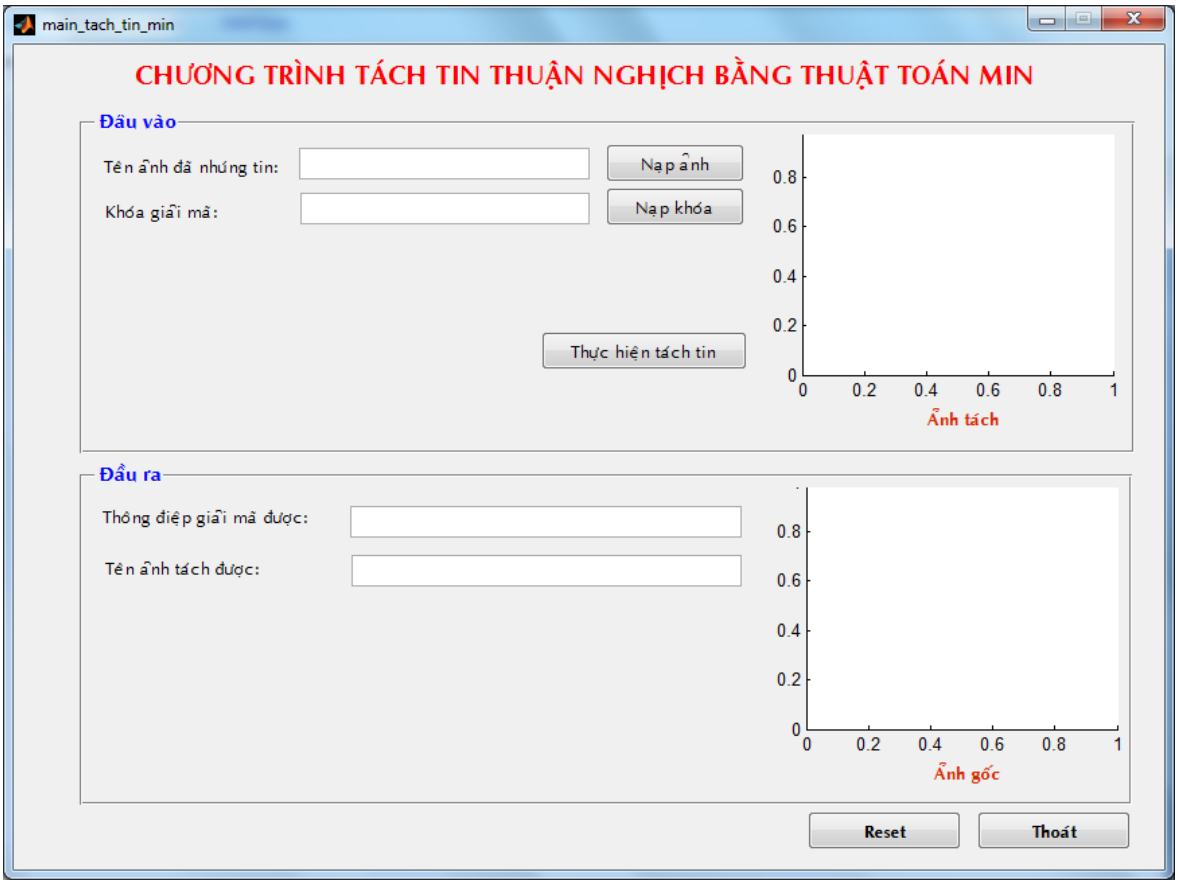
Hình 3.5 Nơi lưu khóa giải mã

Đến đây sẽ có 1 thông báo “da NHUNG xong VAN BAN vao ANH GOC”.



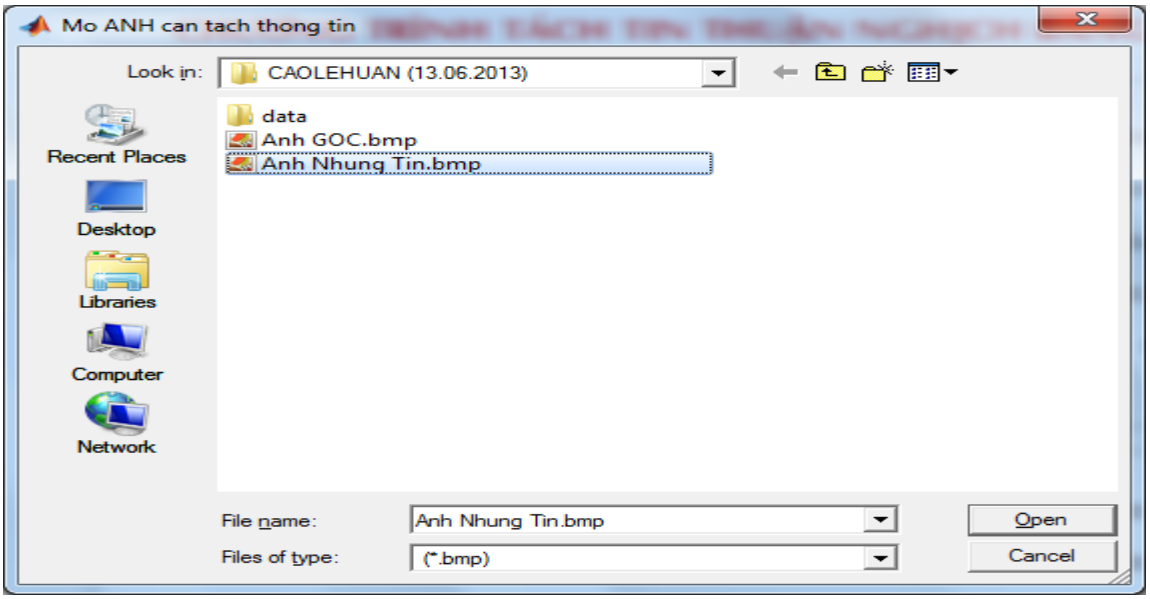
Hình 3.6 Giấu xong tin

Quay lại giao diện chính của chương trình, từ menu "Tach tin" chon "Thuat toan Min" sẽ gọi đến giao diện :



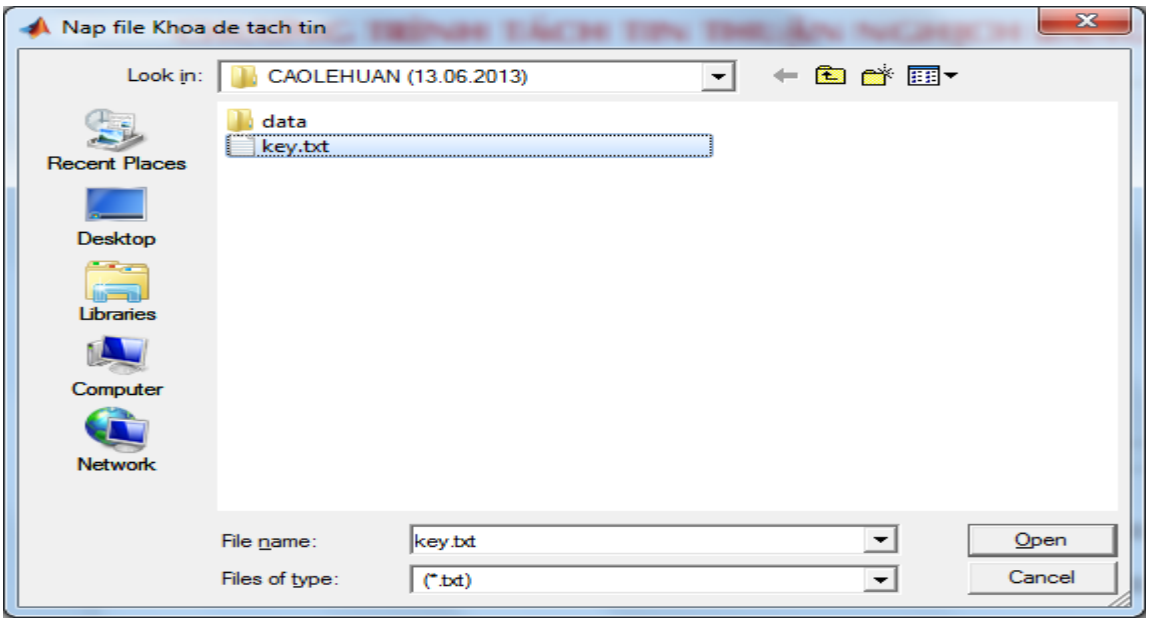
Hình 3.7 Giao diện tách tin Min

Kích vào nút "Nạp ảnh" sẽ xuất hiện hộp thoại để mở ảnh đã giấu tin:



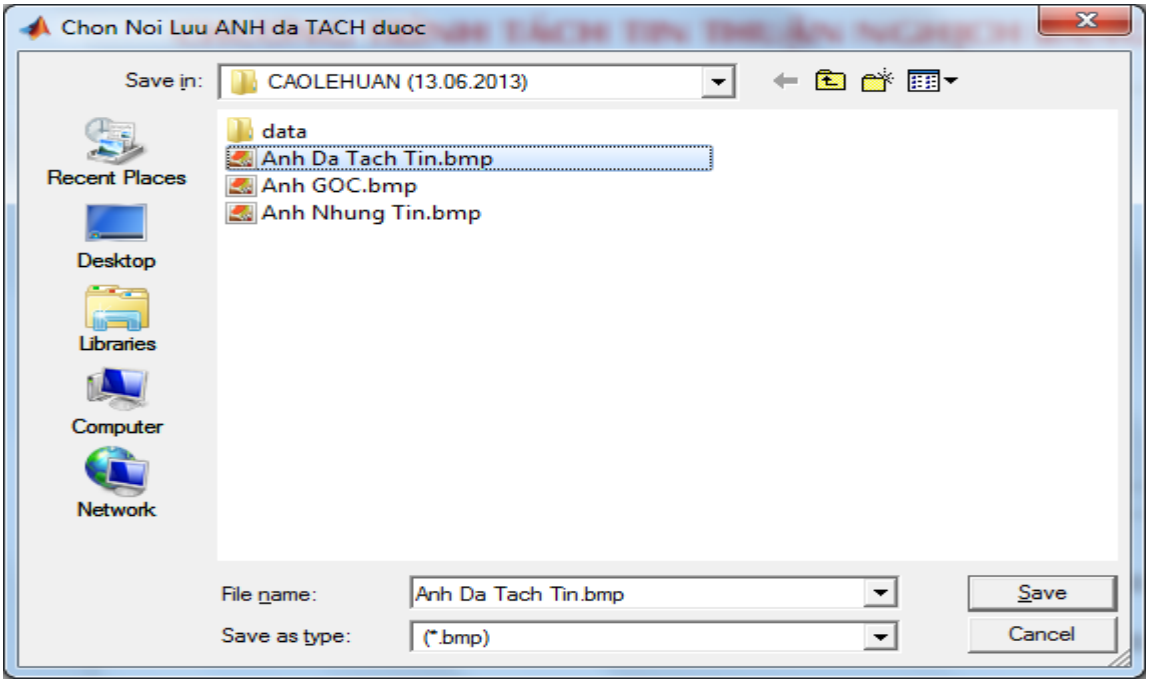
Hình 3.8 Mở ảnh cần tách thông tin

Sau khi mở ảnh đã nhúng tin sẽ có 1 thông báo "Da NAP xong ANH can tach thong tin". Kích vào nút "Nạp khóa" sẽ mở ra hộp thoại để chọn file khóa cần tách tin:



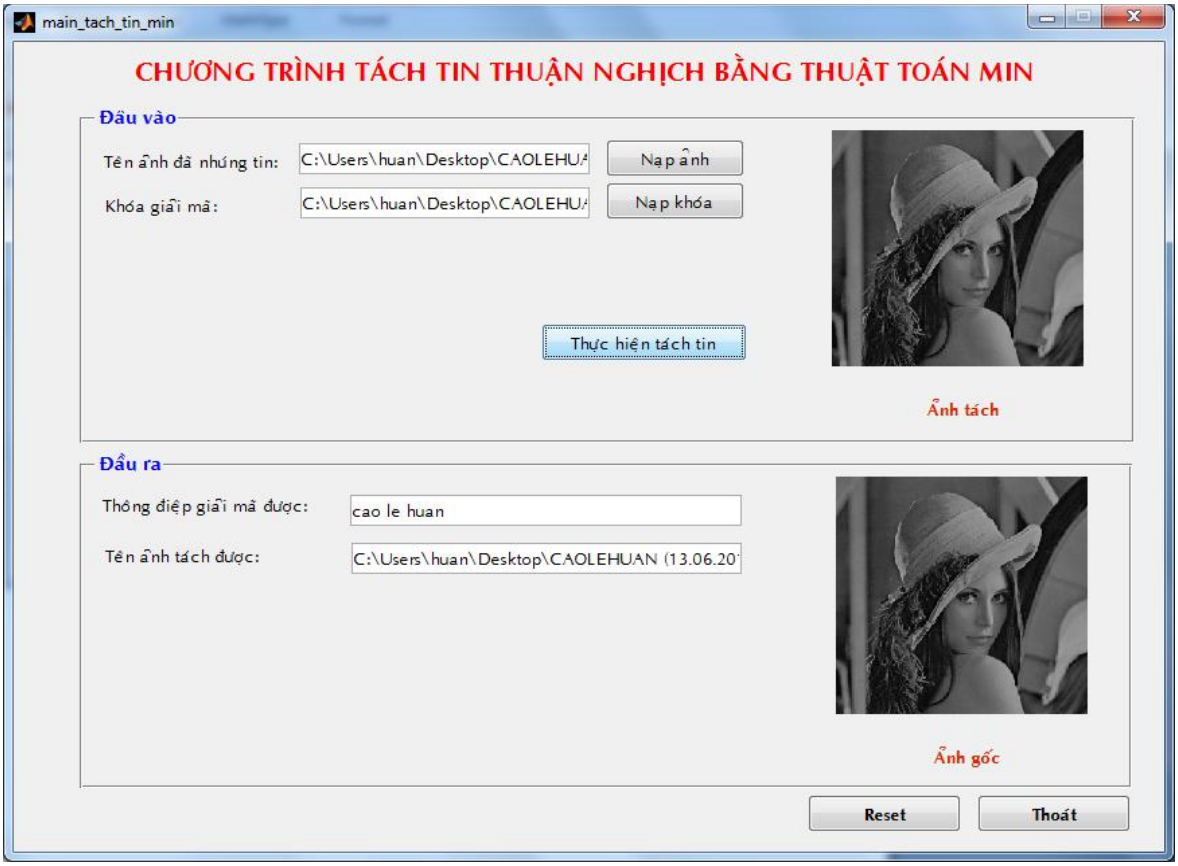
Hình 3.9 Nạp file khóa để tách tin

Đến phần tách tin, ta kích vào nút "Thực hiện tách tin" sẽ xuất hiện hộp thoại chọn nơi lưu ảnh đã tách tin:



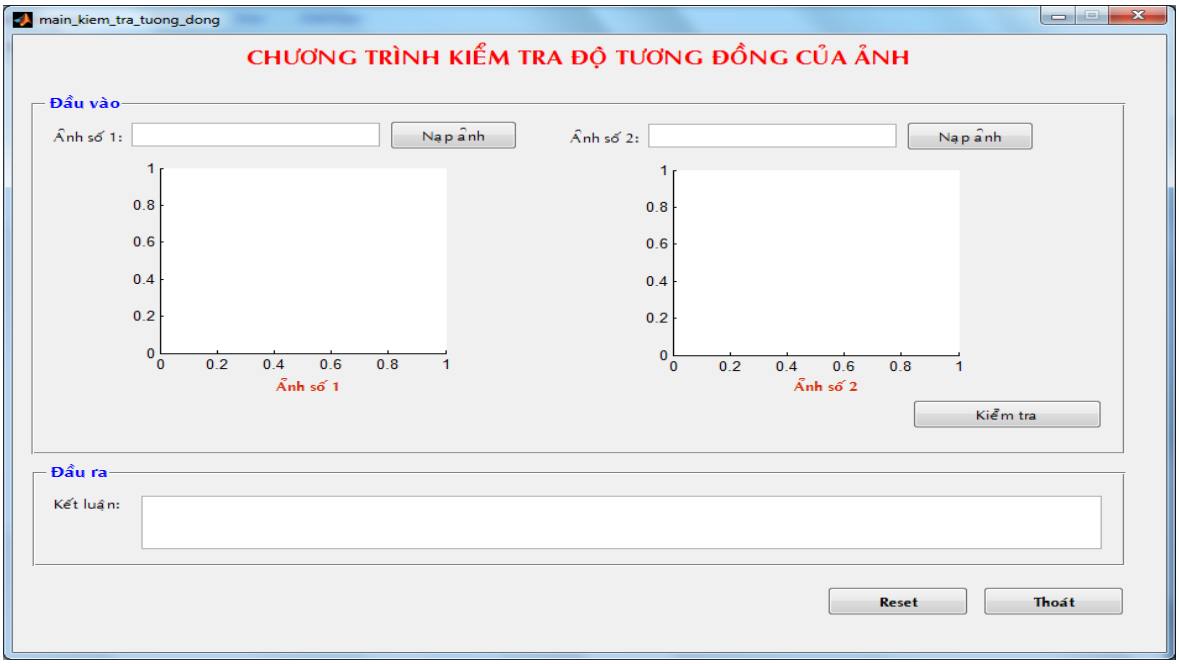
Hình 3.10 Nơi lưu ảnh đã tách tin

Cuối cùng ta nhận đươc thông điệp và ảnh gốc:



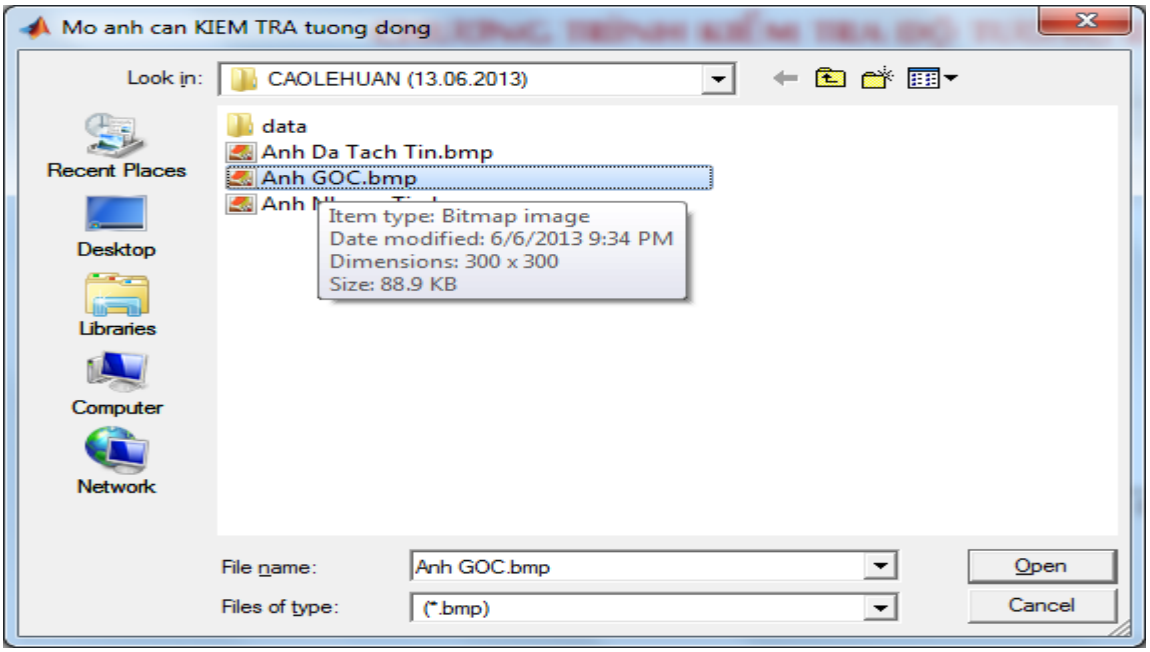
Hình 3.11 Tách xong tin

Trong giao diện chính của chương trình có phần kiểm tra ảnh trước và sau khi giấu tin. Trong giao diện chính kích vào "Kiem tra" sẽ gọi đến giao diện:



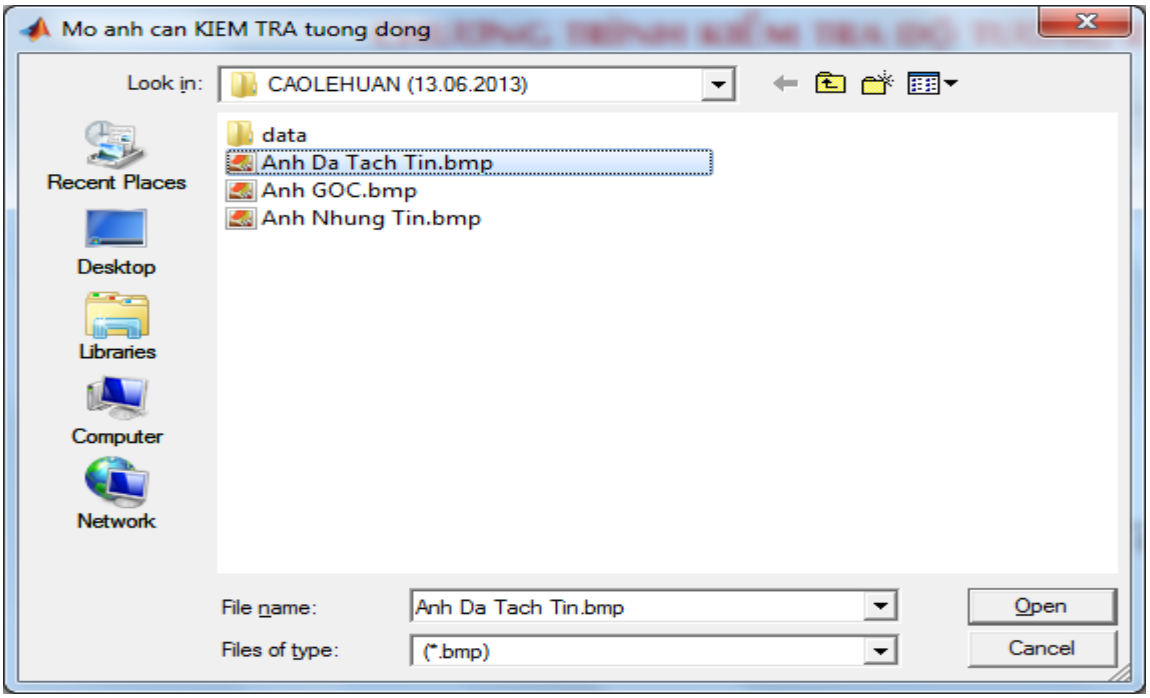
Hình 3.12 Giao diện kiểm tra độ tương đồng của ảnh

Chọn nút "Nạp ảnh " để nhập ảnh số 1. Xuất hiện hộp thoại để chọn ảnh Gốc trước khi giấu tin:



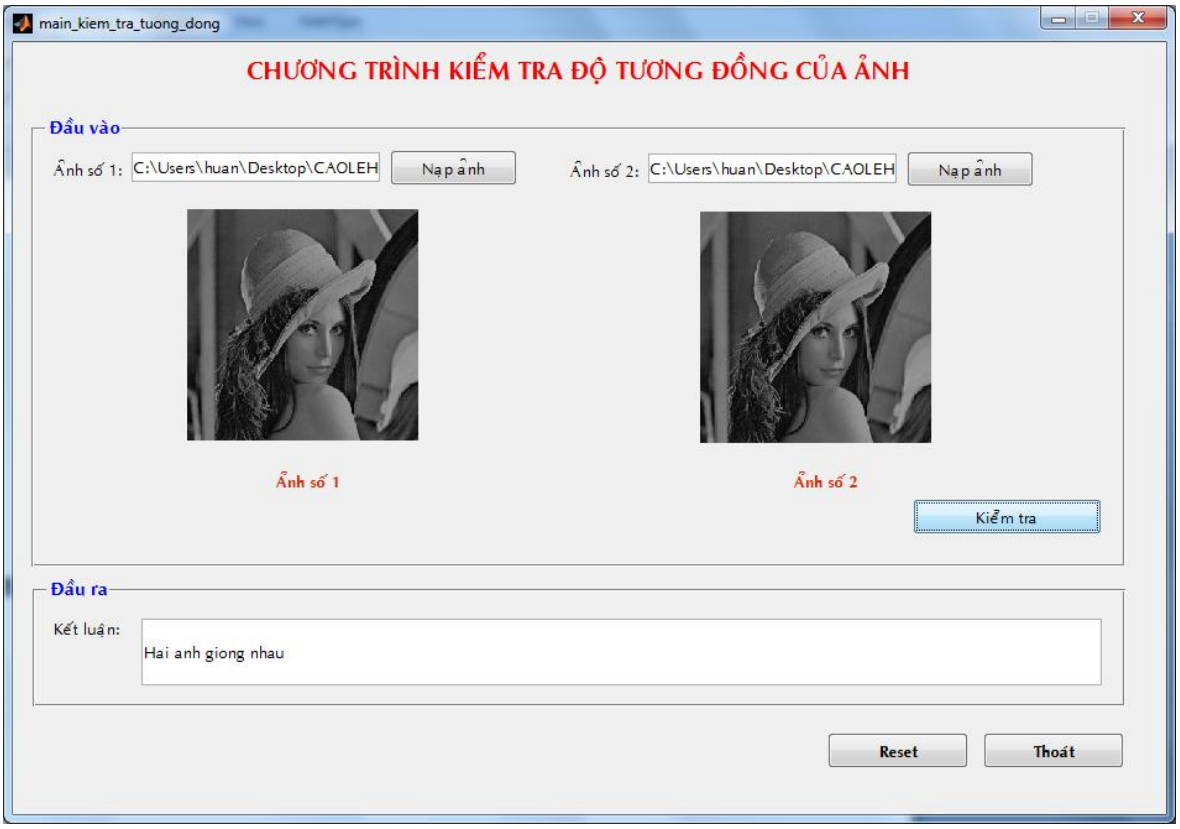
Hình 3.13 Mở ảnh số 1

Chọn nút "Nạp ảnh " để nhập ảnh số 2. Xuất hiện hộp thoại để chọn ảnh Đã Tách Tin:



Hình 3.14 Mở ảnh số 2

Thực hiện kiểm tra bằng cách kích vào nút "Kiểm tra". Kết quả đưa ra sẽ được ghi trong ô kết luận:



Hình 3.15 Kiểm tra xong độ tương đồng của ảnh

Muốn Reset hoặc Thoát thì kích vào nút "Reset" hoặc "Thoát".

Phần Giấu tin và tách tin của thuật toán MAX cũng tương tự các bước như của thuật toán MIN đã đưa ra trên đây.

3.3. KẾT QUẢ THỬ NGHIỆM VÀ NHẬN XÉT

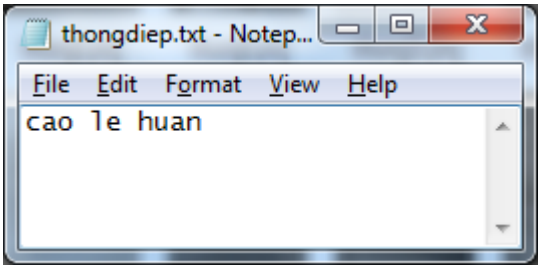
3.3.1. Kết quả thực nghiệm.

Thực nghiệm này sẽ đưa ra khả năng giấu tin khi sử dụng kỹ thuật giấu tin thuận nghịch sử dụng thuật toán **MAXMIN**.

- **Thuật toán MAX.**

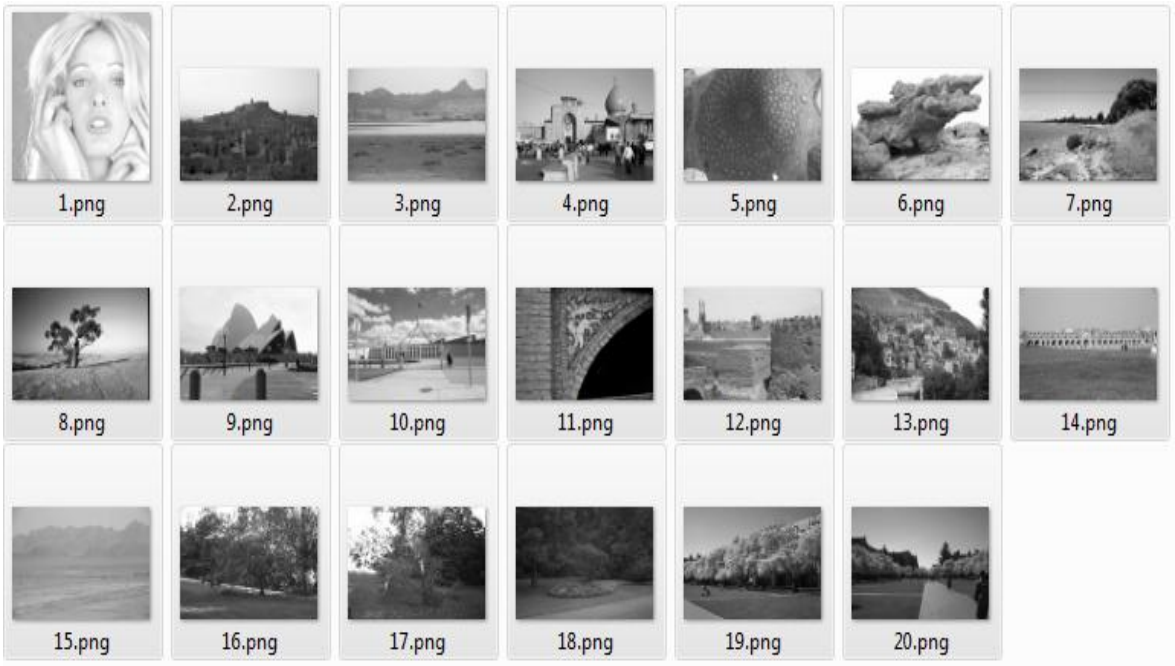
Tập ảnh thử nghiệm bao gồm 20 ảnh PNG cấp xám 8 bit (Hình 3.17).

- **TH1.** Giấu ít thông điệp

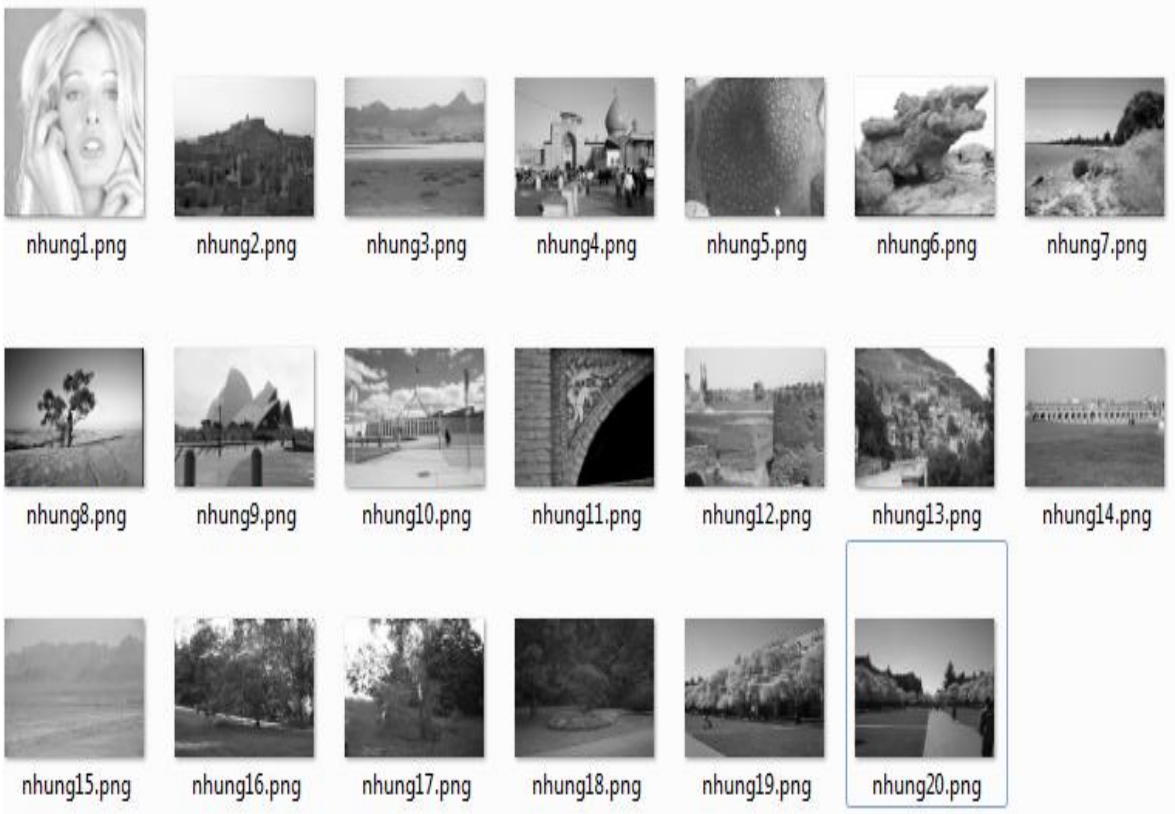


Hình 3.16 Chuỗi thông điệp cần giấu

Thực hiện bằng thuật toán **MAX** giấu thông tin có độ dài 11 bit (Hình 3.16) ta được tập ảnh đã giấu tin (Hình 3.18).



Hình 3.17 Ảnh trước khi giấu tin (TH1, MAX)



Hình 3.18 Ảnh sau khi giấu tin (TH1, MAX)

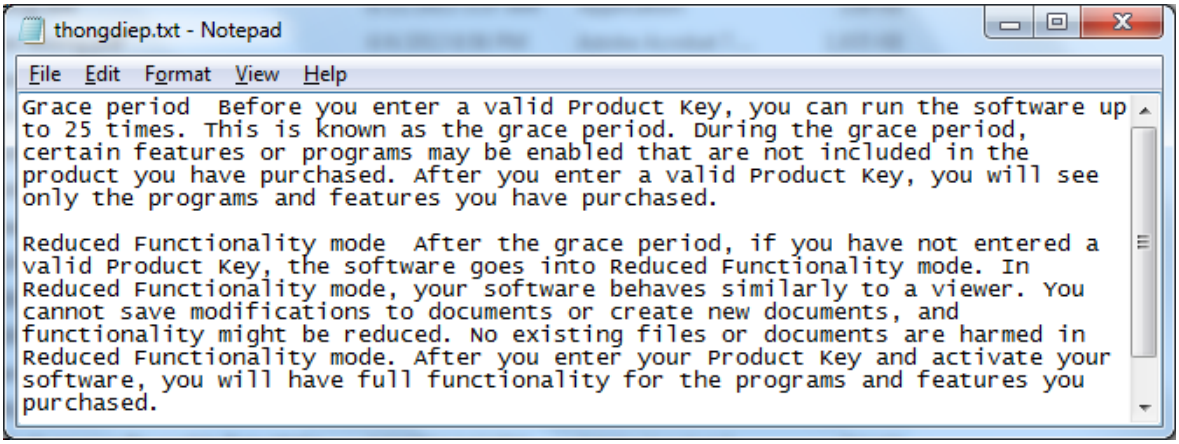
Thực hiện đánh giá PSNR giữa tập ảnh ban đầu với tập ảnh đã giấu tin bằng thuật toán **MAX** ta được kết quả theo Bảng 3.1

Bảng 3.1 Kết quả đánh giá PSNR. (TH1, MAX)

Tên ảnh(kích cỡ ảnh)	**Đánh giá PSNR**(dB)
1.png(512x512)	59.8306 dB
2.png(768x512)	88.0637 dB
3.png(768x512)	76.1539 dB
4.png(768x512)	82.5508 dB
5.png(768x512)	62.9333 dB
6.png(768x512)	87.4495 dB
7.png(768x512)	77.1398 dB
8.png(768x512)	79.6211 dB
9.png(768x512)	82.8714 dB

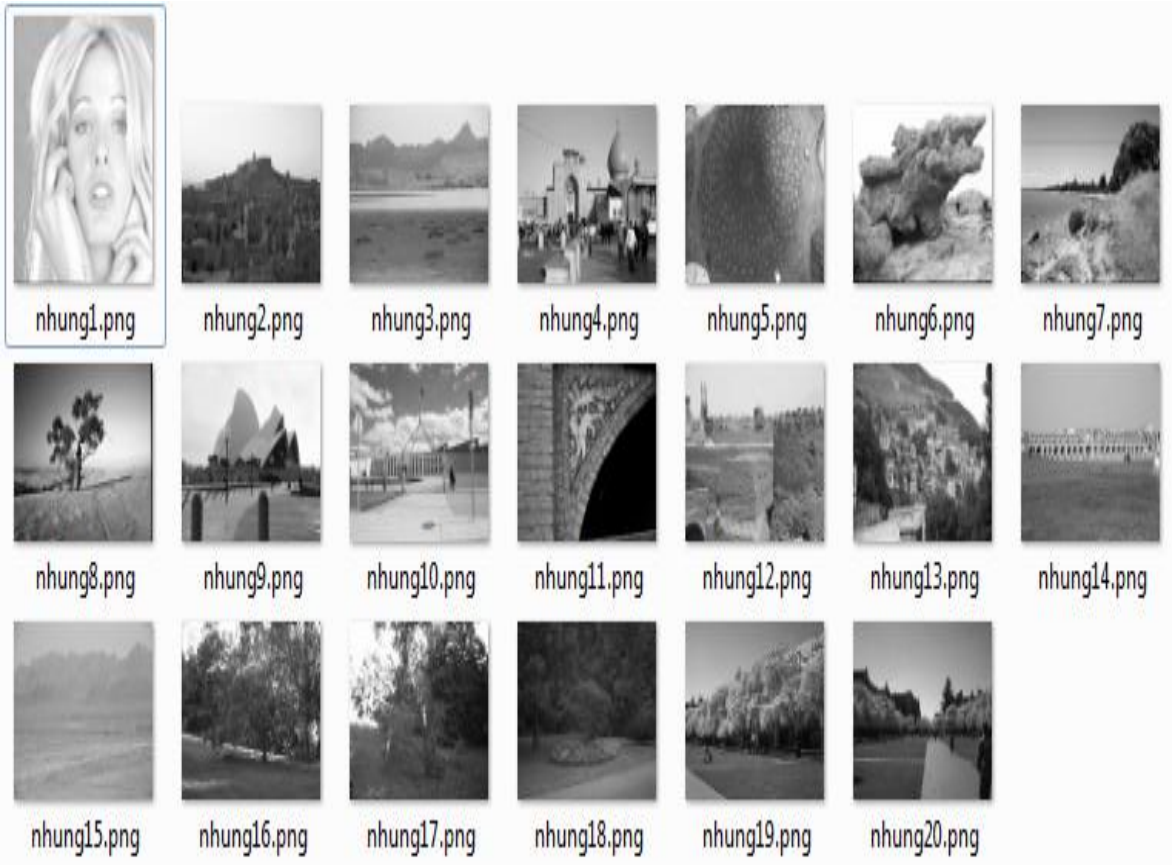
10.png(768x512)	69.6901 dB
11.png(768x512)	59.6002 dB
12.png(768x512)	85.1008 dB
13.png(768x512)	70.824 dB
14.png(768x512)	81.5633 dB
15.png(768x512)	77.076 dB
16.png(756x504)	63.3296 dB
17.png(756x504)	84.4464 dB
18.png(756x504)	76.5843 dB
19.png(756x504)	58.4885 dB
20.png(756x504)	58.4218 dB
Giá trị trung bình	**74.0869 dB**

- **TH2:** Giấu nhiều thông điệp.



Hình 3.19 Chuỗi thông điệp 12000 ký tự cần giấu

Thực hiện bằng thuật toán **MAX** giấu thông tin có độ dài 12000 bit (Hình 3.19) ta được tập ảnh đã giấu tin (Hình 3.20).



Hình 3.20 Ảnh sau khi giấu tin (TH2, MAX)

Thực hiện đánh giá PSNR giữa tập ảnh ban đầu với tập ảnh đã giấu tin bằng thuật toán **MAX** ta được kết quả theo Bảng 3.2

Bảng 3.2 Kết quả đánh giá PSNR (TH2, MAX)

Tên ảnh(kích cỡ ảnh)	**Đánh giá PSNR**(dB)
1.png(512x512)	36.8487 dB
2.png(768x512)	61.355 dB
3.png(768x512)	63.8827 dB
4.png(768x512)	60.7468 dB
5.png(768x512)	36.9066 dB
6.png(768x512)	67.7922 dB

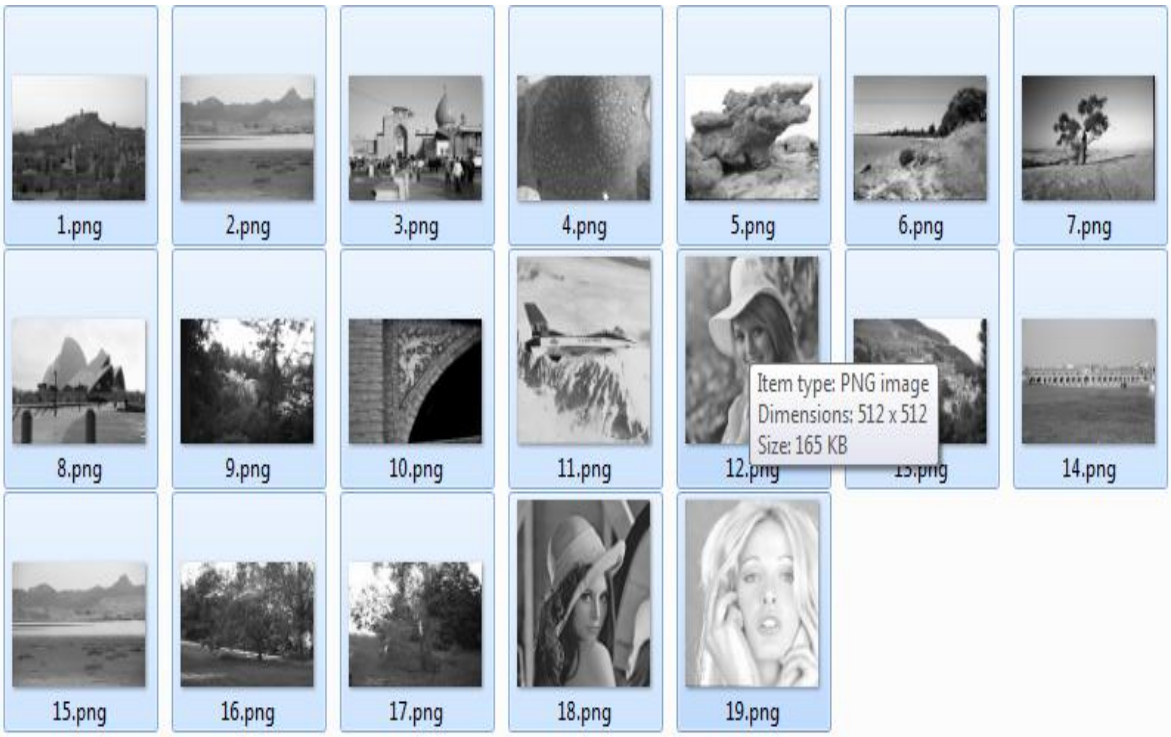
7.png(768x512)	59.6761 dB
8.png(768x512)	59.3074 dB
9.png(768x512)	66.8237 dB
10.png(768x512)	43.8855 dB
11.png(768x512)	41.5657 dB
12.png(768x512)	66.3429 dB
13.png(768x512)	41.4736 dB
14.png(768x512)	61.5856 dB
15.png(768x512)	61.0713 dB
16.png(756x504)	32.8343 dB
17.png(756x504)	32.3698 dB
18.png(756x504)	44.2772 dB
19.png(756x504)	54.9287 dB
20.png(756x504)	54.6931 dB
Giá trị trung bình	**52.4183 dB**

- **Thuật toán MIN**

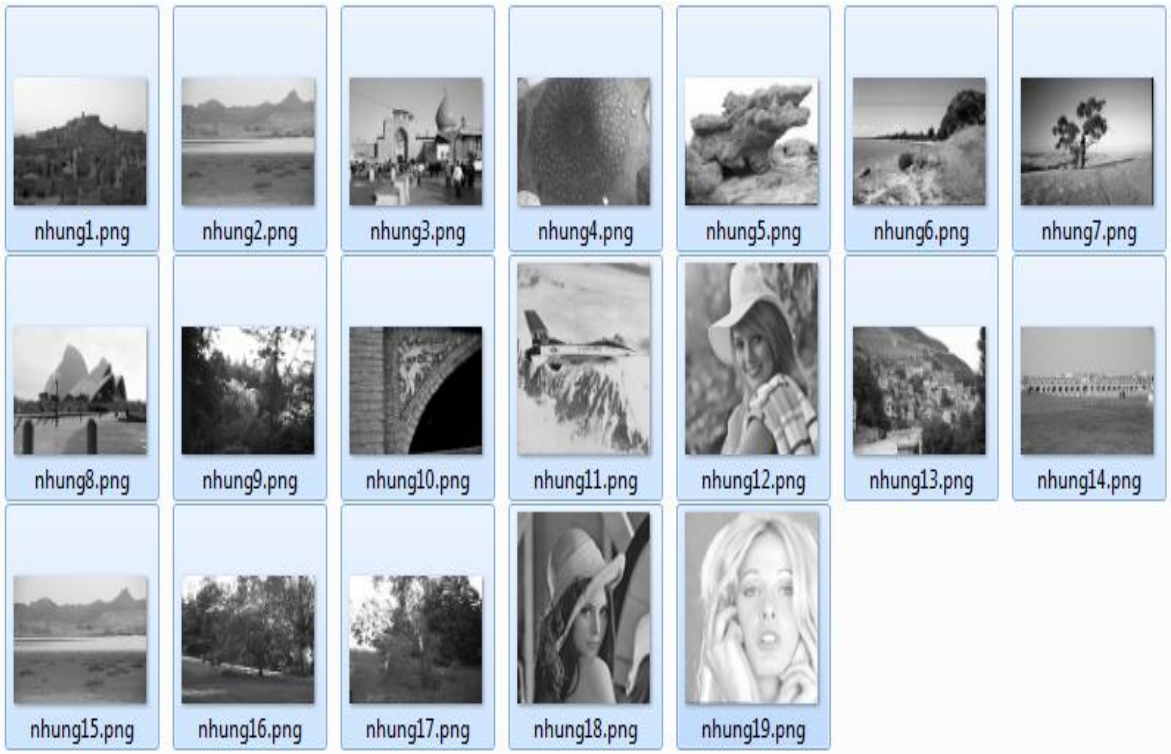
Tập ảnh thử nghiệm bao gồm 19 ảnh PNG cấp xám 8 bit (hình 3.21).

- **TH1.** Giấu ít thông điệp

Thực hiện bằng thuật toán **MIN** giấu thông tin có độ dài 11 bit (Hình 3.16) ta được tập ảnh đã giấu tin (Hình 3.22).



Hình 3.21 Ảnh trước khi giấu tin (TH1, MIN)



Hình 3.22 Ảnh sau khi giấu tin (TH1, MIN)

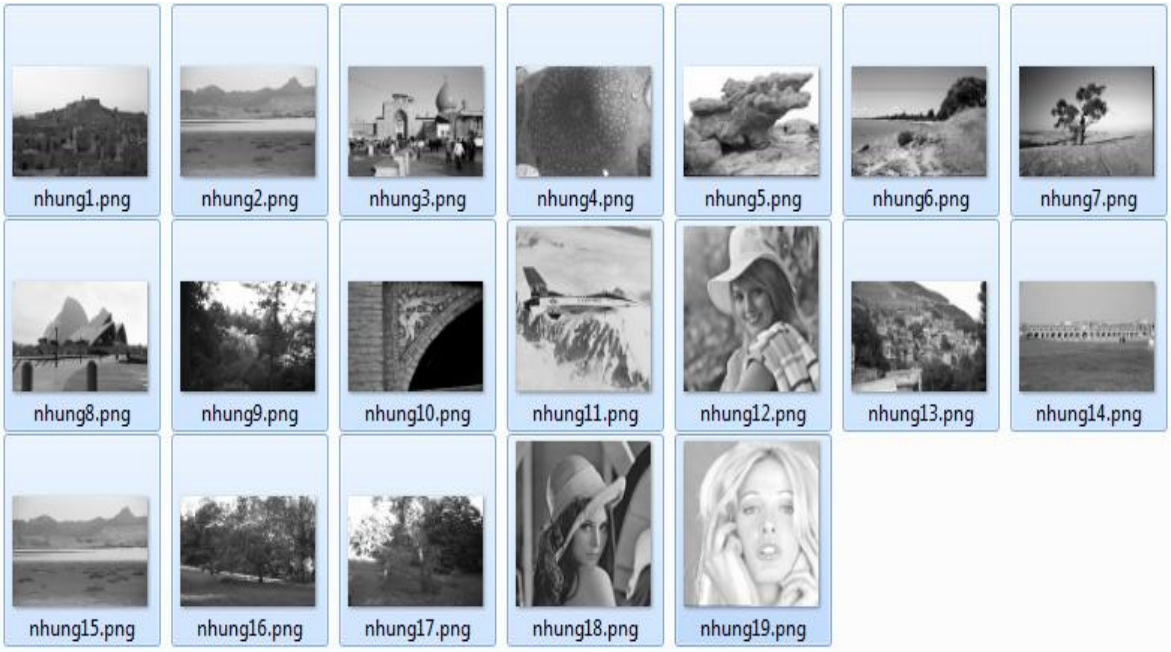
Thực hiện đánh giá PSNR giữa tập ảnh ban đầu với tập ảnh đã giấu tin bằng thuật toán **MIN** ta được kết quả theo Bảng 3.3

Bảng 3.3 Kết quả đánh giá PSNR. (TH1, MIN)

Tên ảnh(kích cỡ ảnh)	**Đánh giá PSNR**(dB)
1.png(768x512)	86.6504 dB
2.png(768x512)	74.3503 dB
3.png(768x512)	78.8316 dB
4.png(768x512)	61.6055 dB
5.png(768x512)	87.4495 dB
6.png(768x512)	75.5645 dB
7.png(768x512)	77.1929 dB
8.png(768x512)	79.5746 dB
9.png(756x504)	77.7808 dB
10.png(768x512)	60.2843 dB
11.png(512x512)	57.1208 dB
12.png(512x512)	80.5621 dB
13.png(768x512)	69.3131 dB
14.png(768x512)	78.951 dB
15.png(768x512)	74.3503 dB
16.png(756x504)	65.0133 dB
17.png(756x504)	84.9094 dB
18.png(512x512)	74.4946 dB
19.png(512x512)	57.164 dB
Giá trị trung bình	**73.7454 dB**

- **TH2:** Giấu nhiều thông điệp.

Thực hiện bằng thuật toán **MIN** giấu thông tin có độ dài 12000 bit ta được tập ảnh đã giấu tin (Hình 3.23).



Hình 3.23 Ảnh sau khi giấu tin (TH2, MIN)

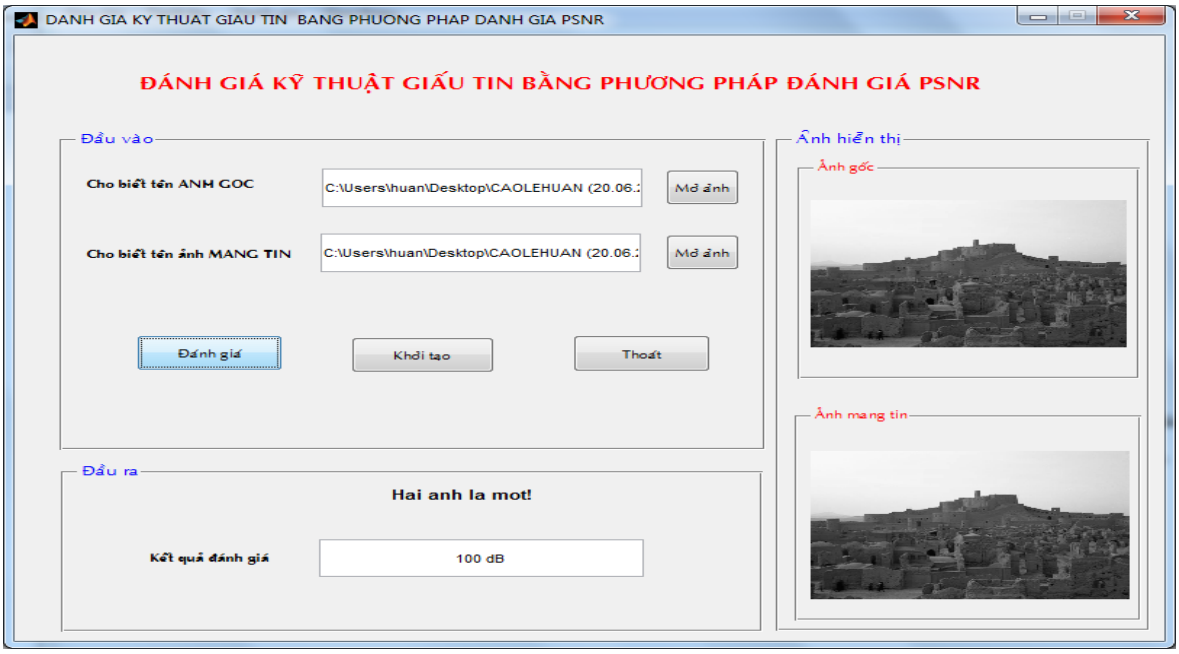
Thực hiện đánh giá PSNR giữa tập ảnh ban đầu với tập ảnh đã giấu tin bằng thuật toán **MIN** ta được kết quả theo Bảng 3.4

Bảng 3.4 Kết quả đánh giá PSNR. (TH2, MIN)

Tên ảnh(kích cỡ ảnh)	**Đánh giá PSNR**(dB)
1.png(768x512)	57.5173 dB
2.png(768x512)	59.8993 dB
3.png(768x512)	57.3846 dB
4.png(768x512)	35.8257 dB
5.png(768x512)	66.1827 dB
6.png(768x512)	57.1663 dB

7.png(768x512)	53.5025 dB
8.png(768x512)	62.1364 dB
9.png(768x512)	35.7386 dB
10.png(768x512)	39.5737 dB
11.png(512x512)	39.5558 dB
12.png(512x512)	39.8695 dB
13.png(768x512)	41.5421 dB
14.png(768x512)	57.7081 dB
15.png(768x512)	59.8993 dB
16.png(756x504)	35.207 dB
17.png(756x504)	33.9323 dB
18.png(512x512)	44.9028 dB
19.png(512x512)	42.9537 dB
Giá trị trung bình	**48.4472 dB**

Sau khi thử nghiệm đánh giá PSNR của tập ảnh trước khi giấu tin và ảnh đã tách tin của cả 2 thuật toán **MAX**, **MIN** đều cho kết quả 100 dB.



Hình 3.24 Đánh giá PSNR ảnh trước khi giấu tin và sau khi tách tin

3.3.2. Nhận xét

Với kết quả thử nghiệm thu được, nếu chuỗi thông điệp nhỏ quan sát bằng mắt thường thì khó có thể phân biệt được ảnh đã giấu và chưa giấu tin, giá trị PSNR trung bình đạt được là khá cao khi giấu tin. Nhưng nếu chuỗi thông điệp lớn giá trị PSNR lại khá thấp. Cụ thể ở thuật toán **MAX**, TH1 giá trị trung bình là 74.0869dB trong khi đó TH2 là 52.4183 dB. Thuật toán **MIN**, TH1 giá trị trung bình là 73.7454 dB trong khi đó TH2 là 48.4472 dB.

Thời gian xử lý giấu tin phụ thuộc lớn vào dữ liệu đầu vào như kích thước ảnh gốc, thông điệp giấu lớn hay nhỏ.

Qua thử nghiệm em nhận thấy kỹ thuật giấu tin sử dụng thuật toán **MAXMIN** có những ưu, nhược điểm sau:

- ✓ Ưu điểm:
  - Khả năng bảo mật cao
  - Giấu được nhiều thông tin
  - Phương pháp này không chỉ hoàn toàn phục hồi môi trường giấu tin mà còn tạo ra một chất lượng nhận diện cao của các hình ảnh được đánh dấu.
  - Hiệu suất tải trọng và PSNR của phương pháp này vượt trội hơn hẳn so với các chương trình hiện có
- ✓ Nhược điểm:
  - Thời gian xử lý giấu tin chậm nếu dữ liệu đầu vào lớn.
  - Phải dùng 2 thuật toán

KẾT LUẬN

Kỹ thuật giấu thông tin trong ảnh là hướng nghiên cứu chính của thuật toán giấu thông tin hiện nay và đã đạt được những kết quả khả quan. Đồ án đã trình bày một số khái niệm liên quan đến kỹ thuật giấu tin thuận nghịch, cũng như trình bày kỹ thuật giấu tin thuận nghịch sử dụng thuật toán **MAXMIN.**

Thuật toán MAXMIN bao gồm hai phần: Thuật toán giấu tin bằng bảo toàn nhỏ nhất và thuật toán giấu tin bằng bảo toàn lớn nhất. Thông thường thuật toán giấu tin bằng bảo toàn nhỏ nhất thì ít hao tổn hơn khi nhúng một thông tin bí mật vào một loạt các hình ảnh. Thuật toán bảo toàn lớn nhất thay thế các thuật toán giấu tin bằng bảo toàn nhỏ nhất khi thuật toán giấu tin bằng bảo toàn nhỏ nhất không có khả năng tiến hành khôi phục lại dữ liệu ẩn trên một hình ảnh nhất định. Để cung cấp một sự lưu trữ ẩn cao hơn và khắc phục các vấn đề vượt ngưỡng, thuật toán **MAXMIN** đã nhúng các bit dữ liệu vào một khối khác biệt mà chúng được tạo ra bằng cách trừ tối thiểu (hoặc tối đa) các giá trị *pixel* từ các điểm ảnh còn lại của khối.

Tuy nhiên, giấu tin mật là vấn đề phức tạp, cộng với khả năng và kinh nghiệm còn hạn chế nên em còn gặp một số khó khăn trong việc tìm hiểu nghiên cứu các kỹ thuật giấu tin thuận nghịch.

Vì vậy em rất mong nhận được sự đóng góp ý kiến quý báu của các thầy cô giáo cũng như bạn bè để báo cáo của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

[1]. Chingyu YANG, High- Peformance Reversible Data Hiding by MinMax Algorithm, Journal of Computational Information Systems 8:1 (2012) 363-370.

[2]. C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel (1999), "Lossless recovery of an original image containing embedded data", *US Patent application*, Docket no: 77102/E-D.

[3]. J. Tian (2002), "Reversible Watermarking by Difference Expansion", *In Proc. of Workshop on Multimedia and Security*, pp. 19-22.

[4]. J. Tian (2002), "Wavelet Based Reversible Watermarking for Authentication", *In Proc. Security and Watermarking of MultimediaContents IV*, Electronic Imaging 2002, Vol. 4675, pp. 679-690.

[5]. Shaowei Weng, Yao Zhao (2008), "A novel reversible data hiding scheme", *International Journal of Invovative Computing, Information and Control*, Vol. 4 (3), pp. 351 – 358.

[6]. Ni, Z., Shi, Y., Ansari, N., Su, W. (2003), "Reversible data hiding", *Proc. ISCAS 2003*, pp. 912–915.

[7]. Sang-Kwang Lee, Young-Ho Suh, and Yo-Sung Ho (2004), "Lossless Data Hiding Based on Histogram Modification of Difference Images", A*dvances in Multimedia Information Processing - PCM 2004*, pp. 340-347.

[8]. J.H. Hwang, J. W. Kim, and J. U. Choi (2006), "A Reversible Watermarking Based on Histogram Shifting", *IWDW 2006*, pp. 384-361.

[9] W. Hong, T.S. Chen, and C.W. Shiu. *Reversible data hiding for high quality images using modificationof prediction error*. The Journal of Systems and software, 82: 1833-1842, 2009. V. Sachnev,H.J. Kim, J. Nam, S. Suresh, and Y.Q. Shi. *Reversible watermarking algorithm using sorting and prediction*. IEEE T. Circuits and Systems for Video Technology, 19 (7): 989-999, 2009.

[10] C.F. Lee, H.L. Chen, and H.K. Tso. *Embedding capacity raising in reversible data hiding based on prediction of different expansion*. The Journal of Systems and Software, 83: 1864-1872, 2010.

[11] K.S. Kim, M.J. Lee, H.Y. Lee, and H.K. Lee. *Reversible data hiding exploiting spatial correlation between sub-sampled images*. Pattern Recognition, 42: 3083-3096, 2009.

[12] H.J. Hwang, H.J. Kim, V. Sachnev, and S.H. Joo. *Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting*. KSII Trans. Internet and Information Systems, 4(4): 655-670, 2010.

[13] Nguyễn Xuân Huy, Trần Quốc Dũng, ***Giáo trình giấu tin và thuỷ vân ảnh***, Trung tâm thông tin tư liệu,TTKHTN - CN 2003

[14] Vũ Trọng Hùng – CT801, "**Kỹ thuật giấu tin thuận nghịch dựa trên miền dữ liệu ảnh"**, tiểu án tốt nghiệp ngành CNTT – 2009.

[15] Đỗ Lâm Hoàng – CT1001, **"Nghiên cứu kỹ thuật giấu tin thuận nghịch trên miền dữ liệu ảnh cấp xám"**, đồ án tốt nghiệp ngành CNTT – 2010.

[16] Trần Đại Dương, **"Kỹ thuật giấu tin thuận nghịch trong ảnh bằng hiệu chỉnh hệ số wavelet"**, đồ án tốt nghiệp ngành CNTT.