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

MỤC LỤC

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 tin mật. Trong đồ án này em đã tìm hiểu kỹ thuật giấu tin trên biên của ảnh nhị phân.

Chương 1. Khái niệm tổng quan: Trình bày tổng quan kỹ thuật giấu tin trong ảnh, cấu trúc ảnh BITMAP và phương pháp đánh giá PSNR (peak signal-to-noise ration) ảnh trước và sau khi giấu tin, kỹ thuật nén ảnh Jpeg.

Chương 2. Kỹ thuật giấu tin trên biên của ảnh nhị phân.

Chương 3. Cài đặt thử nghiệm: Trình bày một số giao diện của chương trình và thử nghiệm kỹ thuật giấu tin trên biên của ảnh nhị phân.

Chương 1. KHÁI NIỆM TỔNG QUAN

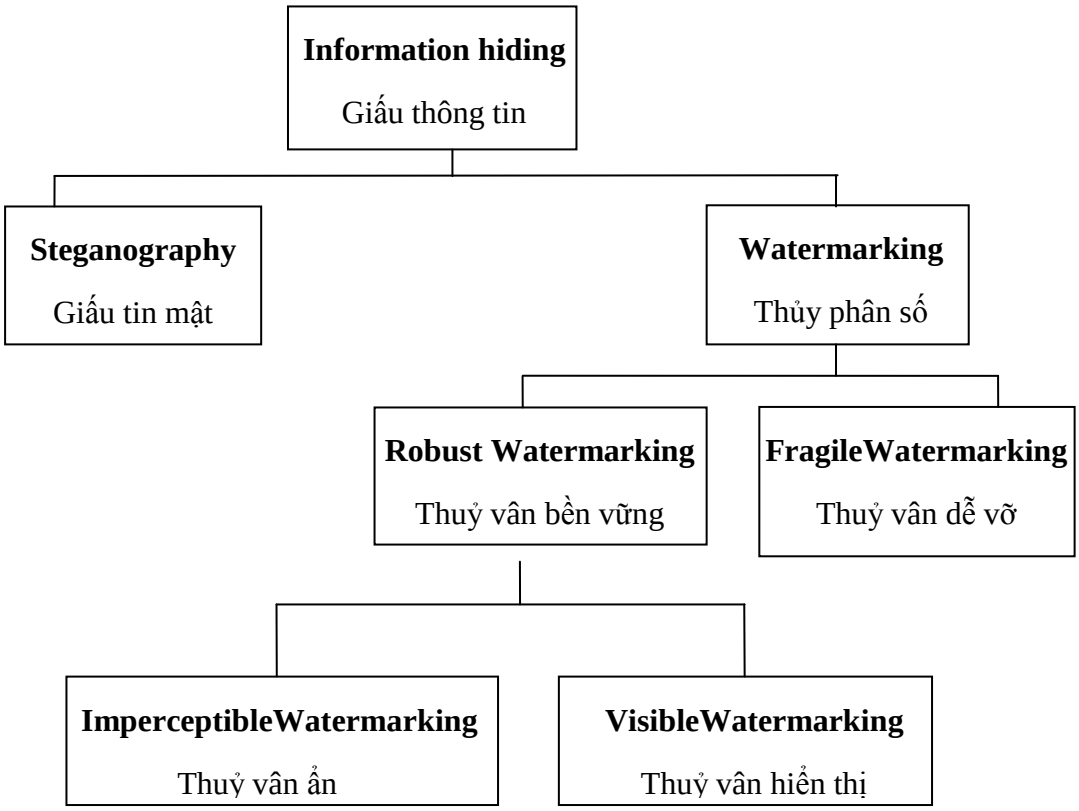
1.1. Tổng quan kỹ thuật giấu tin trong ảnh

1.1.1. Khái niệm

- Giấu tin là kỹ thuật nhúng (giấu) một lượng thông tin nào đó vào trong một đối tượng dữ liệu số khác.
- Giấu tin số là giấu những thông tin số vào trong một đối tượng dữ liệu số khác (gọi là môi trường dấu tin) sao cho môi trường trước và sau khi giấu tin gần như không có sự khác biệt, đồng thời có thể khôi phục lại chính xác các thông tin đã giấu.

1.1.2. Phân loại kỹ thuật giấu tin

- Có thể chia kỹ thuật giấu tin ra làm 2 loại lớn đó là thủy vân (watermarking) và giấu tin mật (steganography).



Hình 1. 1. Sơ đồ phân loại kỹ thuật giấu tin.

- **Thủy vân số (Watermarking):** giấu mẩu tin ngắn, nhưng đòi hỏi độ bền vững cao của thông tin cần giấu (trước các biến đổi thông thường của tệp dữ liệu môi trường).

- Thủy vân bền vững: thường được ứng dụng trong bảo vệ bản quyền. Thủy vân được nhứng trong sản phẩm như một ứng dụng trong bảo vệ bản quyền. trong trường hợp này, thủy vân phải tồn tại bền vững cùng với sản phẩm nhằm chống việc tẩy xóa, làm giả hay biến đổi phá hủy thủy vân.
- Thủy vân dễ vỡ: là kỹ thuật nhúng thủy vân vào trong một đối tượng (sản phẩm) sao cho khi phân bố sản phẩm (trong môi trường mở) nếu có bất kỳ phép biến đổi nào làm thay đổi sản phẩm gốc thì thủy vân đã được giấu trong đối tượng sẽ không còn nguyên vẹn như trước khi dấu.
- Thủy vân ẩn: Cũng giống như giấu tin, bằng mắt thường không thể nhìn được thủy vân ẩn.
- Thủy vân hiện: là loại thủy vân hiện ngay trên sản phẩm và mọi người đều có thể nhìn thấy được.

❖ **Giấu tin mật (Steganography):** Che giấu bản tin (đòi hỏi độ mật cao và dung lượng càng lớn càng tốt) vào môi trường (đối tượng) gốc.

Bảng 1. 1. So sánh giữa giấu tin mật và thủy vân số

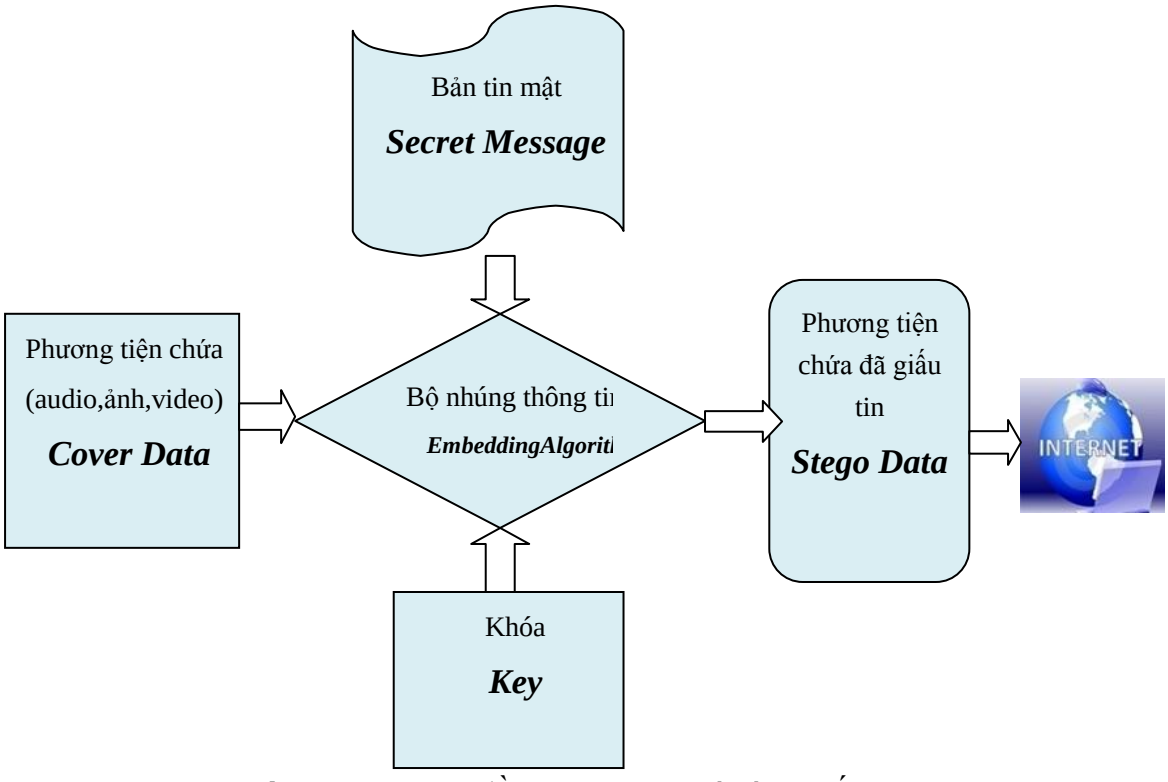
Giấu tin mật	Thủy vân số
- Tập trung vào việc giấu được càng nhiều tin càng tốt, ứng dụng trong truyền dữ liệu mật. - Cố gắng làm ảnh hưởng ít nhất đến chất lượng của đối tượng gốc để không bị chú ý đến dữ liệu đã được giấu trong đó. - Thay đổi đối tượng gốc cũng làm cho dữ liệu giấu bị sai lệch (ứng dụng trong xác thực thông tin). - Bảo mật cho dữ liệu cần giấu. Khía cạnh này tập trung vào kỹ thuật giấu tin mật, tức là giấu tin sao cho giấu được nhiều và người khác khó phát hiện ra thông tin được giấu trong đó.	- Không cần giấu nhiều thông tin, chỉ cần lượng thông tin nhỏ đặc trưng cho bản quyền của người sở hữu. - Trong trường hợp thủy vân nhìn thấy thì thủy vân sẽ hiện ra. - Thủy vân phải bền vững với mọi tấn công có chủ đích hoặc không có chủ đích vào sản phẩm. - Thủy vân số đánh dấu vào chính đối tượng, nhằm khẳng định bản quyền sở hữu hay phát hiện xuyên tạc thông tin.

1.1.3. Mô hình kỹ thuật giấu tin và tách tin cơ bản

Các thành phần chính của một hệ giấu tin và tách tin trong ảnh số gồm:

- *Bản tin mật (Secret Message):* có thể là văn bản hoặc tệp ảnh hay bất kỳ một tệp nhị phân nào, vì quá trình xử lý đều chuyển chúng thành chuỗi các bit.
- *Ảnh phủ (hay ảnh gốc) (Cover Data):* là ảnh được dùng để làm môi trường nhúng tin mật.
- *Khoá bí mật K (Key):* khoá bí mật tham gia vào quá trình giấu tin để tăng tính bảo mật.
- *Bộ nhúng thông tin (Embedding Algorithm):* Những chương trình, thuật toán nhúng tin.
- *Ảnh mang (Stego Data):* là ảnh sau khi đã nhúng tin mật vào đó.
- *Kiểm định (Control) :* kiểm tra thông tin sau khi được giải mã.

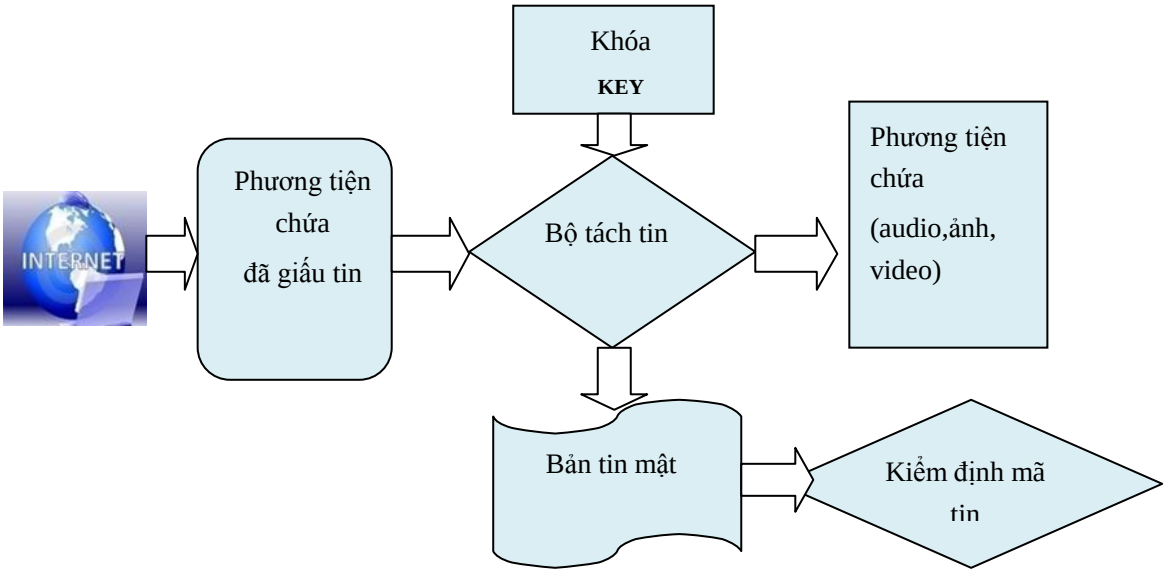
Mô hình của kỹ thuật giấu tin và tách tin cơ bản được mô tả như sau:



Hình 1. 2. Lược đồ chung cho quá trình giấu tin.

- Hình 1. 2 biểu diễn quá trình giấu tin cơ bản. Phương tiện chứa bao gồm các đối tượng được dùng làm môi trường giấu tin như: text, audio, video, ảnh, bản tin mật là một lượng thông tin mang một ý nghĩa nào đó như ảnh, logo, đoạn văn bản… tuỳ thuộc vào mục đích của người sử dụng. Thông tin sẽ được giấu vào

trong phương tiện chứa nhờ một bộ nhúng, bộ nhúng là những chương trình, triển khai các thuật toán để giấu tin và được thực hiện với một khoá bí mật giống như các hệ mật mã cổ điển. Sau khi giấu tin, ta thu được phương tiện chứa bản tin đã giấu và phân phối sử dụng trên mạng.



Hình 1. 3. Lược đồ chung cho quá trình tách tin.

- Hình 1.3 mô tả việc tách thông tin đã giấu. Sau khi nhận được đối tượng phương tiện chứa có giấu thông tin, quá trình tách tin được thực hiện thông qua bộ nhúng thông tin cùng với khoá của quá trình nhúng. Kết quả thu được gồm phương tiện chứa gốc và bản tin mật đã được giấu. Bước tiếp theo bản tin mật thu được sẽ được xử lý kiểm định so sánh với thông tin giấu ban đầu.

1.1.4. Các đặc tính của giấu tin trong ảnh

- *Tính ẩn (tính vô hình):* Khi quan sát ảnh mang bằng mắt thường không phát hiện được thông tin giấu và không gây nghi ngờ cho người xem.Tính ẩn phụ thuộc vào mức độ biến đổi của ảnh mang so với ảnh gốc trong quá trình giấu tin.
- *Tính bền vững:* Ảnh mang có thể phải chịu một tác động nào đó từ bên ngoài như lọc ảnh, làm sắc nét,… dẫn đến mẩu tin tách được M’ # M. Tỉ lệ M’/ M thể hiện tính bền vững của thuật toán giấu tin.
- *Dung lượng dấu tin:* Là tỉ lệ giữa số byte tối đa thông tin có thể giấu được so với kích thước của file ảnh (tính bằng byte). Cùng một thuật toán giấu tin với các file ảnh khác nhau có thể cho tỉ lệ khác nhau. Thông thường các phương pháp giấu tin trong ảnh đều cố làm tăng dung lượng giấu tin, tuy nhiên việc tăng dung lượng giấu tin sẽ ảnh hưởng tới các đặc tính ẩn và tính bền vững.

- *Tính an toàn:* Là khả năng chống lại sự tấn công hoặc giả mạo từ bên ngoài. Một hệ giấu tin tốt phải đảm bảo bí mật không bị tấn công một cách có chủ đích trên cơ sở những hiểu biết về phương pháp giấu tin như có ảnh mang, có ảnh mang và ảnh gốc, có bộ giải mã (nhưng chưa có khóa), ….
- *Độ phức tạp tính toán :* Chủ yếu tính bằng các phép toán thực hiện trong việc giấu tin và giải mã (tách tin). Yêu cầu về độ phức tạp tính toán tùy thuộc từng ứng dụng

1.1.5. Môi trường giấu tin

1.1.5.1. Giấu tin trong ảnh

Hiện nay giấu thông tin trong ảnh là một bộ phận chiếm tỷ lệ lớn trong các chương trình ứng dụng, các phần mềm, hệ thống giấu tin trong đa phương tiện bởi lượng thông tin được trao đổi bằng ảnh là rất lớn và hơn nữa giấu thông tin trong ảnh cũng đóng vai trò hết sức quan trọng trong hầu hết các ứng dụng bảo vệ an toàn thông tin như: xác định xuyên tạc thông tin, bảo vệ quyền tác giả…Thông tin sẽ được giấu cùng dữ liệu ảnh nhưng chất lượng ảnh ít thay đổi và chẳng ai biết được đằng sau ảnh đó mang những thông tin có ý nhĩa. Ngày nay khi ảnh số được sử dụng rất phổ biến thì giấu thông tin trong ảnh đã mang lại nhiều những ứng dụng quan trọng trên các lĩnh vực đời sống xã hội. Ví dụ như các nước phát triển chữ ký tay đã được số hóa và lưu trữ sử dụng như là hồ sơ cá nhân của các dịch vụ ngân hàng tài chính. Phần mềm WinWord của Microsoft cũng cho phép người dung lưu trữ chứ ký trong ảnh nhị phân rồi gắn vào vị trí nào đó trong tệp văn bản để đảm bảo tính an toàn của thông tin.

1.1.5.2. Giấu tin trong audio

Giấu thông tin trong audio mang những đặc điểm riêng khác với giấu thông tin trong các đối tượng đa phương tiện khác. Một trong những yêu cầu cơ bản của giấu thông tin là đảm bảo tính chất ẩn của thông tin được giấu đồng thời không làm ảnh hưởng tới chất lượng của dữ liệu. Để đảm bảo yêu cầu này ta lưu ý rằng kỹ thuậtgiấu thông tin trong ảnh phụ thuộc vào hệ thống thị giác của con người – HSV (Human Vision System) còn kỹ thuật giấu thông tin trong audio lại hệ phục thuộc vào hệ thống tính giác HAS (Human Auditory System). Một vấn đề khó khăn ở đây là hệ thống thính giác của con người nghe được các tín hiệu ở các giải tần rộng và công suất lớn nên đã gây khó dễ đối với các phương pháp giấu tin trong audio. Nhưng tai con người lại kém trong việc phát hiện sự khác biệt của các giải tần và

công suất, có nghĩa là các âm thanh to, cao tần có thể che giấu được các âm thanh nhỏ thấp một cách dễ dàng.

Vấn đề khó khăn thứ hai đối với giấu tin trong audio là kênh truyền tin, kênh truyền hay băng thông chậm sẽ ảnh hưởng tới chất lượng thông tin sau khi giấu. Giấu thông tin trong audio đòi hỏi yêu cầu rất cao về tính đồng bộ và tính an toàn của thông tin. Các phương pháp giấu thông tin trong audio đều lợi dụng điểm yếu trong hệ thống thính giác của con người.

1.1.5.3. Giấu tin trong video

Cũng giống như giấu thông tin trong ảnh hay audio, giấu tin trong video cũng được quan tâm và phát triển mạnh mẽ cho nhiều ứng dụng như điều khiển truy cập thông tin, nhận 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. Nhiều nhà nghiên cứu đã dùng những hàm cosin riêng và các hệ số truyền sóng riêng để giấu tin. Trong các thuật toán khởi nguồn thì thường các kỹ thuật cho phép giấu các ảnh vào trong video nhưng thời gian gần đây các kỹ thuật cho phép giấu cả âm thanh hình ảnh vào video.

1.1.5.4. Giấu tin trong dạng văn bản 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 hóa 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 gì dữ liệu đa phương tiện như ảnh, video, audio. Gần đây đã có một số nghiên cứu giấu tin trong cơ sở dữ liệu quân hệ, các gói IP truyền trên mạng, chắc chắn sau này còn phát triển tiếp cho các môi trường dữ liệu số khác.

1.1.6. Ứng dụng kỹ thuật giấu tin trong ảnh

- *Liên lạc bí mật :* Giấu tin trong ảnh rồi gửi đi trên mạng ít gây sự chú ý hơn so với sử dụng mật mã. Ngoài ra việc sử dụng công nghệ mã hóa có thể bị hạn chế và cấm sử dụng. Có thể dùng để liên lạc bí mật trong cả thương mại để gửi đi một bí mật thương mại trong quân sự, ngoại giao để gửi đi một bản vẽ hay một thông điệp nhạy cảm.

- *Bảo vệ bản quyền tác gi:* Một thông tin nào đó mang ý nghĩa quyền sở hữu tác giả (ví dụ như logo của công ty) được bí mật nhúng vào trong các sản phẩm để xác nhận quyền sở hữu khi bán hoặc phân phối sản phẩm, thêm vào đó có thể gán một nhãn thời gan để chống giả mạo.
- *Điểm chỉ số:* Thủy vân được sử dụng để nhận diện người gửi hay người nhận của một thông tin nào đó trong ứng dụng phân phối sản phẩm. Dùng để xác định người nhận sản phẩm, về mặt nào đó có ý nghĩa như số se-ri sản phẩm.
- *Gán nhãn :* Các chú giải, minh họa có thể được nhúng vào trong ảnh, khi đó nếu sao chép thông thường thì thông tin nhúng cũng được sao chép và chỉ có chủ sở hữu hoặc người được cấp phép mới có thể tách ra được các chú giải này.
- *Điều khiển truy nhập:* Các thiết bị phát hiện thủy vân (ở đây sử dụng phương pháp phát hiện thủy vân đã giấu mà không cần thông tin gốc) được gán sẵn vào trong các hệ thống đọc ghi, tùy thuộc vào việc có thủy vân hay không để điều khiển (cho phép / cấm) truy cập. Ví dụ như hệ thống quản lý sao chép DVD đã được ứng dụng ở Nhật.

1.1.7. Tính chất, đặc trưng của giấu tin trong ảnh

1.1.7.1 Phương tiện chứa có dữ liệu tri giác tĩnh

Dữ liệu gốc ở đây là dữ liệu ảnh tĩnh, dù đã giấu thông tin vào trong ảnh hay chưa, thì khi người xem ảnh bằng thị giác, dữ liệu ảnh không thay đổi theo thời gian. Khác với dữ liệu audio hay video, khi xem hay nghe, thì dữ liệu gốc sẽ thay đổi liên tục với tri giác của con người theo các đoạn hay các bài , các ảnh,…

1.1.7.2 Giấu tin phụ thuộc ảnh

Kỹ thuật giấu tin phụ thuộc vào các loại ảnh khác nhau. Chẳng hạn đối với ảnh đen trắng, ảnh xám hay ảnh màu, ta có những kỹ thuật riêng do các loại ảnh với đặc trưng khác nhau. Ảnh nén và ảnh không nén cũng áp dụng những kỹ thuật giấu tin khác nhau, vì ảnh nén có thể làm mất thông tin khi nén ảnh ....

1.1.7.3 Giấu tin lợi dụng khả năng thị giác con người

Giấu tin trong ảnh cũng gây ra những thay đổi trên dữ liệu ảnh gốc. Dữ liệu ảnh được quan sát bằng hệ thống thị giác con người, nên các kỹ thuật giấu tin phải đảm bảo yêu cầu gây ra những th bản là những thay đổi trên ảnh phải rất nhỏ, sao cho bằng mắt thường không thể nhận ra được sự thay đổi đó, vì có như thế thì mới đảm bảo được độ an toàn cho thông tin giấu.

1.1.7.4 Giấu tin không làm thay đổi kích thước ảnh

Các phép toán giấu tin sẽ được thực hiện trên dữ liệu của ảnh. Dữ liệu ảnh bao gồm cả phần header (là nơi lưu thông tin về tệp, kích thước, và địa chỉ offset về vùng dữ liệu), bảng màu (có thể có) và dữ liệu ảnh. Khi giấu tin, các phương pháp giấu đều biến đổi giá trị của các bít trong dữ liệu ảnh trước hay sau khi giấu tin, là như nhau

1.1.7.5 Đảm bảo chất lượng ảnh sau khi giấu tin

Đây là yêu cầu quan trọng đối với giấu tin trong ảnh. Sau khi giấu tin bên trong, ảnh phải đảm bảo yêu cầu không bị biến đổi, để có thể không bị phát hiện dễ dàng so với ảnh gốc.

1.1.8 Các hướng tiếp cận của giấu tin trong ảnh

1.1.8.1 tiếp cận trên miền không gian của ảnh

Đây là hướng tiếp cận cơ bản và tự nhiên trong số các kỹ thuật giấu tin. Miền không gian ảnh là miền dữ liệu ảnh gốc, tác động lên miền không gian ảnh chính là tác động lên các điểm ảnh, thay đổi trực tiếp giá trị các điểm ảnh. Đây là hướng tiếp cận tự nhiên, bởi vì khi nói đến việc giấu tin trong ảnh người ta thường nghĩ ngay đến việc thay đổi các điểm ảnh nguồn. Một phương pháp phổ biến của hướng tiếp cận này là phương pháp tác động đến bít ít quan trọng nhất của mỗi điểm ảnh.

Ý tưởng cơ bản của phương pháp tác động đến bít ít quan trọng nhất (LSB – Least Significant Bit) của các điểm ảnh là chọn ra từ mỗi điểm ảnh các bít ít có ý nghĩa nhất về mặt tri giác, để sử dụng cho việc giâu tin. Việc bit nào được coi là ít tri giác nhất và bao nhiêu bít có thể được lấy ra để thay thế đều phụ thuộc vào khả năng hệ thống thị giác của con người và nhu cầu về chất lượng ảnh trong các úng dụng.

1.1.8.2 Tiếp cận trên miền tần số của ảnh

Trong một số trường hợp cách khảo sát trực tiếp ở trên cũng gặp phải khó khăn nhất dịnh hoặc rất phức tập và hiệu quả không cao, do đó ta có thể dùng phương pháp khảo sát gián tiếp thông qua các kỹ thuật biến đổi. Các biến đổi này làm nhiệm vụ chuyển miền biến số độc lập sang miền khác, và như vậy tín hiệu và hệ thống rời rạc sẽ được biểu diễn trong miền mới với các biến số mới.

Mỗi cách biến đổi sẽ có những thuận lợi riêng, tỳ từng trường hợp mà sử dụng biến đổi nào. Sau khi khảo sát, biến đổi xong các tín hiệu và hệ thông rời rạc trong miền các biến số mới này, nếu cần thiết có thể dùng các biến đổi ngược để đưa chúng về miền biến số độc lập.

Phương pháp khảo sát gián tiếp sẽ làm đơn giản rất nhiều các công việc gặp phải khi dùng phương pháp khảo sát trực tiếp trong miền biến số độc lập tự nhiên. Có nhiều phép biến đổi, trong đó phổ biến là biến đổi Fourier DFT, biến đổi Cosin rời rạc DCT, biến đổi sóng nhỏ DWT…

1.2. Cấu trúc ảnh BITMAP

Bảng 1. 2. Cấu trúc ảnh bitmap.

Bitmap Header (54 byte)
Color Palette
Bitmap Data

Mỗi file ảnh Bitmap gồm 3 phần theobảng sau:

1.2.1. Bitmap header

- Thành phần bitcount (Bảng 1. 3 Thông tin về Bitmap header) của cấu trúc Bitmap header cho biết số bit dành cho mỗi điểm ảnh và số lượng màu lớn nhất của ảnh.

Bảng 1. 3. 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	

1.2.2. Palette màu

- Bảng màu của ảnh, chỉ những ảnh nhỏ hơn hoặc bằng 8 bit mới có bảng màu.

Bảng 1. 4. 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

1.2.3. Ảnh nhị phân

Ảnh nhị phân là ảnh kỹ thuật số mà chỉ có hai giá trị có thể cho mỗi pixel. Thông thường hai màu sắc được sử dụng cho một ảnh nhị phân là hai màu đen và trắng mặc dù có thể sử dụng bất kì hai màu sắc khác. Các màu sắc được sử dụng cho đối tượng trong hình là màu nền trước khi phần còn lại của hình ảnh là màu nền.

Ảnh nhị phân được gọi là nhị cấp hoặc hai cấp. Điều này có nghĩa là mỗi điểm ảnh được lưu giữ như là một bit (0 hoặc 1).

Ứng dụng chính của ảnh nhị phân được dùng theo tính logic để phân biệt đối tượng ảnh với nền hay để phân biệt điểm biên với điểm khác.

Ảnh nhị phân thường được lưu trữ trong bộ nhớ như là một ảnh bitmap, một mảng đóng gói của các bit.

Ảnh nhị phân được lưu trữ như là một ảnh định dạng bitmap hay ảnh định dạng IMG.

Sự đơn giản của định dạng tệp tin BMP, và sự phổ biến của nó trong windows và các hệ điều hành khác, cũng như thực tế là định dạng này cũng tương đối tốt, làm cho nó trở thành một định dạng hình ảnh rất phổ biến, chương trình xủ lí từ nhiều hệ điều hành có thể đọc và viết.

Bảng 1. 5. Cấu trúc ảnh bitmap của ảnh nhị phân.

Header(1)
Info header(2)
Optional palette (3)
IMAGE DATA(4)

(1). BITMAPFILEHEADER(14 byte): là phần chứa các thông tin về kiểu ảnh, kích thước, độ phân giải, số bit dùng cho một pixel, cách mã hóa, vị trí bảng màu …

(2). BITMAPINFOHEADER: là nơi lưu trữ thông tin chi tiết về các hình ảnh bitmap, mà sẽ được sử dụng để hiển thị hình ảnh trên màn hình.

(3). OPTINAL PALETE: là một khối byte (một bảng) danh sách các màu có sẵn để sử dụng trong chỉ mục màu sắc cụ thể của ảnh.

(4). IMAGE DATA: là nơi lưu trữ mô tả dữ liệu của ảnh. Điểm ảnh được lưu trữ "ngược lại" đối với hình ảnh bình thường bằng raster, bắt đầu ở góc trái bên dưới, từ trái sang phải, và sau đó liên tiếp bởi hàng từ đáy lên đỉnh của hình ảnh.

1.3. Phương pháp đánh giá PSNR(peak signal-to-noise ratio)

PSNR là phương pháp đánh giá độ nhiễu của ảnh trước và sau khi giấu tin, đơn vị đo là logarithm decibel. Thông thường PSNR càng cao thì độ nhiễu của ảnh trước và sau khi giấu tin càng thấp. Giá trị PSNR được coi là tốt ở vào khoảng 35dB và nhỏ hơn 20dB là không chấp nhận được. Hiện nay PSNR được dùng rộng rãi trong kỹ thuật đánh giá chất lượng hình ảnh và video.

Cách đơn giản nhất là định nghĩa thông qua trung bình lỗi bình phương (MSE – mean squared error) được dùng cho ảnh 2 chiều có kích thước $m \times n$ trong đó I và K là ảnh gốc và ảnh được khôi phục tương ứng:

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

PSNR được định nghĩa bởi:

$$\text{PSNR} = 10*\log_{10}\left(\frac{MAX_1^2}{MSE}\right) = 20*\log_{10}\left(\frac{MAX_1}{\sqrt{MSE}}\right)$$

Ở đây, MAX(I) là giá trị tối đa của điểm ảnh trên ảnh I. Khi các điểm ảnh được biểu diễn bởi 8 bit, thì giá trị của nó là 255. Trường hợp tổng quát, điểm ảnh được biểu diễn bởi B bit, MAX(I) là 2^B-1. Với ảnh màu biểu diễn 3 giá trị RGB trên 1 điểm ảnh, các tính toán cho PSNR tương tự ngoại trừ việc tính MSE là tổng của 3 giá trị (tính trên 3 kênh màu RGB) chia cho kích thước của ảnh và chia cho 3.

Với ảnh nhị phân các điểm ảnh trên ảnh nhị phân được biểu diễn bởi 2 bit 0 hoặc 1, nên giá trị của MAX(I) = 1.

1.4 Kỹ thuật nén ảnh JPEG

Một tính chất chung nhất của tất cả các ảnh số đó là tương quan giữa các pixel ở cạnh nhau lớn, điều này dẫn đến dư thừa thông tin để biểu diễn ảnh. Dư thừa thông tin sẽ làm cho việc mã hoá không tối ưu. Do đó công việc cần làm để nén ảnh là phải tìm được các biểu diễn ảnh với tương quan nhỏ nhất để giảm thiểu độ dư thừa thông tin của ảnh. Thực tế, có hai kiểu dư thừa thông tin được phân loại như sau:

- *Dư thừa trong miền không gian*: tương quan giữa các giá trị pixel của ảnh, điều này có nghĩa rằng các pixel lân cận của ảnh có giá trị gần giống nhau (trừ những pixel ở giáp đường biên ảnh).

- *Dư thừa trong miền tần số*: Tương quan giữa các mặt phẳng màu hoặc dải phổ khác nhau.

Trọng tâm của các nghiên cứu về nén ảnh là tìm cách giảm số bit cần để biểu diễn ảnh bằng việc loại bỏ dư thừa trong miền không gian và miền tần số càng nhiều càng tốt.

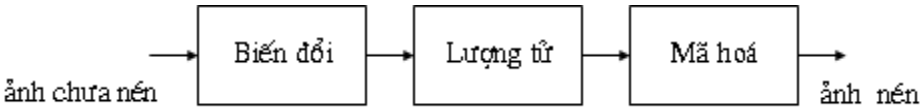
1.4.1 Các kỹ thuật nén ảnh được sử dụng

- Nén ảnh không mất thông tin : với phương pháp này sau khi giải nén ta khôi phục được chính xác ảnh gốc. Các phương pháp nén này bao gồm mã hoá Huffman, mã hoá thuật toán…

- Nén ảnh có mất thông tin: ảnh giải nén có một sự sai khác nhỏ so với ảnh gốc. Các phương pháp này bao gồm:

  - Lượng tử hoá vô hướng: PCM và DPCM
  - Lượng tử hoá vector

- Mã hoá biến đổi: biến đổi cosin rời rạc (DCT), biến đổi Fourier nhanh (FFT)
- Mã hoá băng con

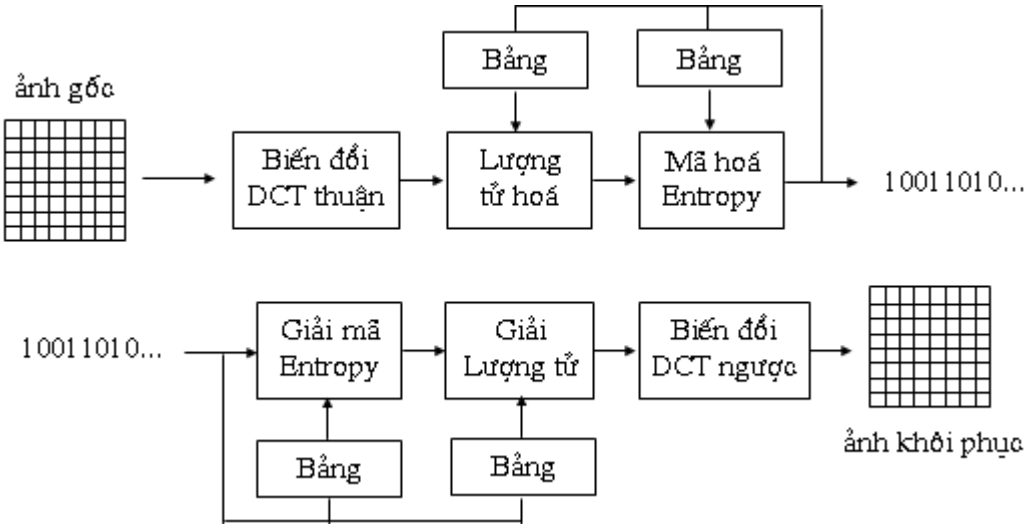


Hình 1.5. Sơ đồ khối một hệ thống nén ảnh điển hình.

Nội dung của đề tài này sẽ thảo luận về phương pháp nén ảnh dùng biến đổi cosin rời rạc DCT (Discrete Cosin Transform): đang được dùng trong chuẩn nén ảnh JPEG hiện nay.

1.4.2 Mã hoá biến đổi DCT

Nguyên tắc chính của phương pháp mã hoá này là biến đổi tập các giá trị pixel của ảnh trong miền không gian sang một tập các giá trị khác trong miền tần số sao cho các hệ số trong tập giá trị mới này có tương quan giữa các điểm ảnh gần nhau nhỏ hơn.



Hình 1.6. Sơ đồ mã hóa và giải mã dùng biến đổi DCT.

1.4.3 Biến đổi DCT thuận và nghịch

Vì ảnh gốc có kích thước rất lớn cho nên trước khi đưa vào biến đổi DCT, ảnh được phân chia thành các khối vuông, mỗi khối này thường có kích thước 8 x 8 pixel và biểu diễn các mức xám của 64 điểm ảnh, các mức xám này là các số nguyên dương có giá trị từ 0 đến 255. Việc phân khối này sẽ làm giảm được một

phần thời gian tính toán các hệ số chung, mặt khác biến đổi cosin đối với các khối nhỏ sẽ làm tăng độ chính xác khi tính toán với dấu phẩy tĩnh, giảm thiểu sai số do làm tròn sinh ra.

Biến đổi DCT là một công đoạn chính trong các phương pháp nén sử dụng biến đổi. 2 công thức ở đây minh hoạ cho 2 phép biến đổi DCT thuận nghịch đối với mỗi khối ảnh có kích thước 8 x 8. Giá trị $x(n_1, n_2)$ biểu diễn các mức xám của ảnh trong miền không gian, $X(k_1, k_2)$ là các hệ số sau biến đổi DCT trong miền tần số.

$$X(k_1,k_2) = \frac{\varepsilon_{k1}\varepsilon_{k2}}{4} \sum_{n1=0}^{7} \sum_{n2=0}^{7} x(n_1,n_2) \cos\frac{(2n_1+1)k_1\pi}{16} \cos\frac{(2n_2+1)k_2\pi}{16} \quad (1.1)$$

$$x(n_1,n_2) = \frac{\varepsilon_{k1}\varepsilon_{k2}}{4} \sum_{n1=0}^{7} \sum_{n2=0}^{7} X(k_1,k_2) \cos\frac{(2n_1+1)k_1\pi}{16} \cos\frac{(2n_2+1)k_2\pi}{16} \quad (1.2)$$

Với $\varepsilon_{k1} = \begin{cases} 1/\sqrt{2} & \text{khi } k_1 = 0 \\ 0 & \text{khi } 1 < k_1 < 8 \end{cases}$ và $\varepsilon_{k2} = \begin{cases} 1/\sqrt{2} & \text{khi } k_2 = 0 \\ 0 & \text{khi } 1 < k_2 < 8 \end{cases}$

Mỗi khối 64 điểm ảnh sau biến đổi DCT thuận sẽ nhận được 64 hệ số thực DCT (bảng 1.6). Mỗi hệ số này có chứa một trong 64 thành phần tần số không gian hai chiều. Hệ số với tần số bằng không theo cả hai hướng (tương ứng với k1 và k2 bằng 0) được gọi là hệ số một chiều DC, hệ số này chính là giá trị trung bình của 64 điểm ảnh trong khối. 63 hệ số còn lại gọi là các hệ số xoay chiều AC. Hệ số một chiều DC tập trung phần lớn năng lượng của ảnh.

Bảng 1.6. Các bước của quá trình mã hóa biến đổi DCT đối với 1 khối.

130	132	132	129	133	133	134	135
135	133	133	131	133	137	138	136
132	134	133	136	139	142	137	137
137	136	136	135	135	136	135	138
139	138	139	135	136	139	137	140
134	136	137	136	134	136	136	143
137	133	138	136	136	136	139	146
139	137	138	134	133	138	140	146

Các giá trị mức xám của 1 khối 64 điểm ảnh

DCT →

1089.1	-10.4	8.1	-2.4	-0.4	-2.2	4.9	0
-12.5	0.6	-5.9	6.3	-2.9	-1.0	-2.8	0.4
-4.9	3.8	3.9	-1.2	-0.5	0.9	-0.3	1.8
-4.2	1.2	0.7	-3.5	0.3	-0.8	0	1.3
0.6	4.2	2.7	-0.4	-1.1	-0.8	0.4	0.8
2.6	-1.2	-2.4	-1.9	2.9	-0.6	-0.5	0.8
1.1	-1.7	-3.1	-0.4	-0.8	-2.5	-0.9	2.3
3.1	-1.7	-2.6	2.0	-0.1	-0.7	0.6	0.5

Các hệ số sau biến đổi DCT thuận

DC

68	1	1	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Các hệ số sau lượng tử hoá

Chú ý rằng bản thân biến đổi DCT không làm mất thông tin vì DCT là một biến đổi tuyến tính chuyển các giá trị của điểm ảnh từ miền không gian thành các hệ số trong miền tần số. Nếu biến đổi DCT thuận và nghịch được tính toán với độ chính xác tuyệt đối và nếu các hệ số DCT không phải qua bước lượng tử và mã hoá thì ảnh thu được sau biến đổi DCT ngược sẽ giống hệt ảnh gốc.

1.4.4 Lượng tử và giải lượng tử

Sau khi thực hiện biến đổi DCT, 64 hệ số sẽ được lượng tử hoá dựa trên một bảng lượng tử gồm 64 phần tử *Q(u,v)* với $0 \le u, v \le 7$. Bảng này được định nghĩa bởi từng ứng dụng cụ thể (hình 1.7 là ví dụ ma trận lượng tử hay sử dụng). Các phần tử trong bảng lượng tử có giá trị từ 1 đến 255 được gọi là các bước nhảy cho các hệ số DCT. Quá trình lượng tử được coi như là việc chia các hệ số DCT cho bước nhảy lượng tử tương ứng, kết quả này sau đó sẽ được làm tròn xuống số nguyên gần nhất. Công thức (3) thể hiện việc lượng tử với $F(u,v)$ là các hệ số DCT, $F^Q(u,v)$ là các hệ số sau lượng tử, các hệ số này sẽ được đưa vào bộ mã hoá Entropy.

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Hình 1.7. Ma trận lượng tử.

$$F^{Q}(u,v) = IntegerRound\left(\frac{F(u,v)}{Q(u,v)}\right) \quad (3)$$

Mục đích của việc lượng tử hoá là giảm số lượng bit cần để lưu trữ các hệ số biến đổi bằng việc giảm độ chính xác của các hệ số này cho nên lượng tử là quá trình xử lý có mất thông tin.

Quá trình giải lượng tử ở phía bộ giải mã được thực hiên ngược lại. Các hệ số sau bộ giải mã entropy sẽ nhân với các bước nhảy trong bảng lượng tử (bảng lượng tử được đặt trong phần header của ảnh JPEG). Kết quả này sau đó sẽ được đưa vào biến đổi DCT ngược.

1.4.5 Mã hóa và giải mã Huffman

Mã hoá là bước cuối cùng trong hệ thống nén ảnh dựa trên biến đổi DCT. Chuẩn nén ảnh JPEG hiện nay dùng phương pháp mã hoá Huffman, đây là phép mã hoá không làm mất thông tin.

Phương pháp mã hoá Huffman là phương pháp dựa vào mô hình thống kê. Dựa vào dữ liệu gốc, người ta tính tần suất xuất hiện của các ký tự. Việc tính tần xuất được thực hiện bằng cách duyệt tuần tự tệp gốc từ đầu đến cuối. Việc xử lý ở đây tính theo bit. Trong phương pháp này, người ta gán cho các ký tự có tần suất cao một từ mã ngắn, các ký tự có tần xuất thấp từ mã dài. Nói một cách khác, các ký tự có tần xuất càng cao được gán mã càng ngắn và ngược lại. Rõ ràng với cách thức này, ta đã làm giảm chiều dài trung bình của từ mã hoá bằng cách dùng chiều dài biến đổi. Tuy nhiên, trong một số tình huống khi tần suất là rất thấp, ta có thể không được lợi một chút nào, thậm chí còn bị thiệt một ít bit.

Thuật toán mã hoá bao gồm 2 bước chính:

-Giai đoạn tính tần suất của các ký tự trong dữ liệu gốc: Duyệt tệp gốc một cách tuần tự từ đầu đến cuối để xây dựng bảng mã. Tiếp sau đó là sắp xếp lại bảng mã theo thứ tự tần suất giảm dần.

-Giai đoạn thứ hai: mã hoá. Duyệt bảng tần suất từ cuối lên đầu để thực hiện ghép 2 phần tử có tần suất thấp nhất thành một phần tử duy nhất. Phần tử này có tần xuất bằng tổng 2 tần suất thành phần. Tiến hành cập nhật lại bảng và đương nhiên loại bỏ 2 phần tử đã xét. Quá trình được lặp lại cho đến khi bảng chỉ có một phần tử. Quá trình này gọi là quá trình tạo cây mã Huffman vì việc tập hợp được tiến hành nhờ một cây nhị phân với 2 nhánh. Phần tử có tần suất thấp ở bên phải, phần tử kia ở bên trái. Với cách tạo cây này, tất cả các bit dữ liệu/ ký tự là nút lá; các nút trong là các nút tổng hợp. Sau khi cây đã tạo xong, người ta tiến hành gán mã cho các nút lá. Việc mã hoá rất đơn giản: mỗi lần xuống bên phải ta thêm 1 bit "1" vào từ mã; mỗi lần xuống bên trái ta thêm 1 bit "0". Tất nhiên có thể làm ngược lại, chỉ có giá trị mã thay đổi còn tổng chiều dài là không đổi. Cũng chính do lý do này mà cây có tên gọi là cây mã Huffman như trên đã gọi.

Quá trình giải nén tiến hành theo chiều ngược lại khá đơn giản. Người ta cũng phải dựa vào bảng mã tạo ra trong giai đoạn nén (bảng này được giữ lại trong cấu trúc đầu của tệp nén cùng với dữ liệu nén). Thí dụ, với một tệp dữ liệu mà tần suất các ký tư cho bởi:

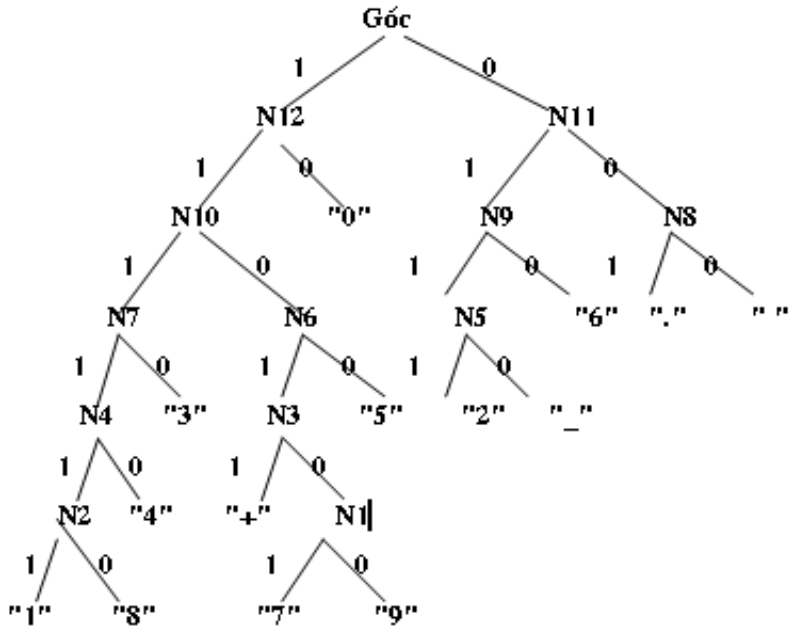
Ký tự	Tần suất	Ký tự	tần suất	xác suất
"1"	152	"0"	1532	0.2770
"2"	323	"6"	602	0.1088
"3"	412	"."	536	0.0969
"4"	226	" "	535	0.0967
"5"	385	"3"	112	0.0746
"6"	602	"5 "	385	0.0696
"7"	92	"2"	323	0.0585
"8"	112	"_"	315	0.0569

"9"	87	"4"	226	0.0409
"0"	1532	"+"	220	0.0396
"."	536	"1"	152	0.0275
"+"	220	"8"	112	0.0203
"_"	315	"7"	92	0.0167
" "	535	"9"	87	0.0158

Bảng tần xuất **Bảng tần suất sắp theo thứ tự giảm dần**

Lưu ý rằng, trong phưng pháp Huffman, mã của ký tự là duy nhất và không mã nào là phần bắt đầu của mã khác. Vì vậy, khi đọc tệp nén từng bit từ đầu đến cuối ta có thể duyệt cây mã cho đến một lá, tức là ký tự đã được giải nén.

Cây mã Hufman tương ứng



Hình 1.8. Cây mã Huffman .

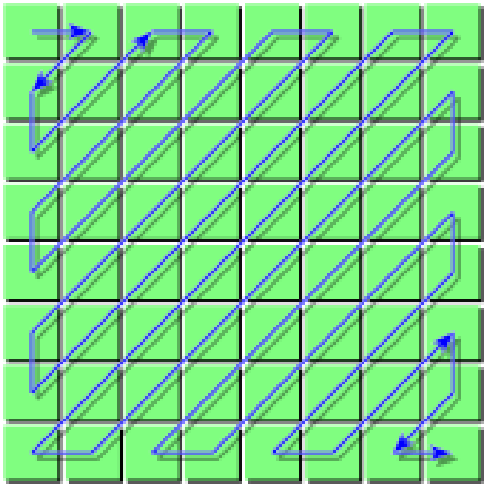
Bảng từ mã gán cho các ký tự bởi mã hoá Huffman

"0"	10	"_"	0110
"6"	010	"4"	11110
"."	001	"+"	11011
" "	000	"1"	111111

"3"	1110	"8"	111110
"5"	1100	"7"	110101
"2"	0111	"9"	110100

Áp dụng phương pháp trên người ta tính tần suất xuất hiện các hệ số. Việc tính tần suất được thực hiện bằng cách duyệt tuần tự từ đầu khối đến cuối khối, sau đó, những hệ số có tần suất cao được gắn cho một từ mã ngắn, các hệ số có tần suất thấp được gán một từ mã dài. Với cách thức này chiều dài trung bình của từ mã đã giảm xuống.

Bảng 1.7. Bảng zigzag của các thành phần ảnh JPEG



Các hệ số thu được sau khi lượng tử hoá sẽ được sắp xếp thành một chuỗi các ký hiệu theo kiểu “zig-zag” (theo đường zig-zag trong bảng 1.7) để đặt các hệ số có tần số thấp lên trước các hệ số tần số cao. Các hệ số này sẽ được mã hoá dựa trên bảng mã Huffman sao cho chiều dài trung bình của từ mã là nhỏ nhất. Bảng mã này cũng sẽ được đặt trong phần mào đầu của ảnh để thực hiện giải nén ảnh.

Chương 2: GIẤU TIN TRÊN ẢNH NHỊ PHÂN

2.1. Giới thiệu về giấu tin trong ảnh nhị phân

Đối tượng làm môi trường chứa tin của thuật toán này là ảnh nhị phân đen trắng dạng bitmap. Ảnh nhị phân đen trắng bao gồm các điểm ảnh chỉ có màu đen hoặc trắng (tương ứng với bit 0 hoặc bit 1). Để giấu dữ liệu, ta sẽ tách ma trận điểm ảnh (pixel) thành ma trận bit F kích thước *mxn* rời nhau, và giấu tin trên mỗi ma trận đó, Bởi vậy các thuật toán chỉ cần quan tâm tới phương pháp giấu dữ liệu trên ma trận F.

Một số thuật toán giấu tin trên ảnh nhị phân nổi tiếng hiện nay như:Wu-Lee[2], CPT[3], CPTE[4]. Các thuật toán này đều dựa trên thao tác biến đổi ma trận bit.

2.2. Một số kỹ thuật giấu tin trên ảnh nhị phân

2.2.1. Giấu tin theo khối bit

Ý tưởng cơ bản của kỹ thuật này là chia ảnh gốc thành các khối nhỏ và trong mỗi khối nhỏ sẽ giấu 1 bit thông tin [1].

Quá trình giấu tin:

- Với một ảnh gốc kích thước MxN, chia phần thông tin ảnh thành các khối nhỏ có kích thước m×n, số các khối nhỏ sẽ là (M×N)/(m×n) khối. Vì ảnh là đen trắng nên mỗi khối là một ma trận hai chiều m dòng, n cột các phần tử có giá trị 0 hoặc 1.
- Chọn các khối chưa giấu tin để thực hiện giấu tin, các khối được chọn cho đến khi giấu hết các thông tin cần giấu hoặc khi đã chọn hết các khối.
- Với mỗi khối ảnh F kích thước m×n và bit đang cần giấu b, tiến hành biến đổi F thành F’ để giấu bit b sao cho: SUM (F’) mod 2 = b
- Như vậy, mỗi lần giấu một bit, có thể xảy ra hai trường hợp: SUM (F) mod 2 = b, khi đó ta giữ nguyên khối ảnh. Ngược lại chọn ngẫu nhiên một bit trong khối F và tiến hành đảo giá trị của bit này để được khối ảnh mới F’.

Quá trình tách tin: Khi nhận được ảnh đã giấu tin, việc tách tin sẽ thực hiện theo các bước:

- Chia ảnh thành các khối có kích thước giống kích thước khối đã sử dụng khi thực hiện giấu, đây chính là khoá để giải mã.

- Với mỗi khối ảnh đã giấu tin F’ được chọn theo thứ tự như quá trình giấu tin, thực hiện tách lấy bit thông tin đã giấu theo công thức: b = SUM (F’) mod 2.
- Như vậy, sau khi xét hết các khối đã giấu, ta thu được một chuỗi bit, chuỗi này là thông tin nhị phân đã giấu cần phải lấy ra.
- Lược đồ giấu tin CB có thể giấu được 1 bit thông tin vào một khối kích thước m×n bit mà chỉ thay đổi tối đa 1 bit trong đó.

2.2.2. Thuật toán Wu-Lee

Là một thuật toán giấu tin khá phổ biến của M. Wu và J. Lee [2]. Trong thuật toán Wu-Lee, môi trường giấu tin là một ảnh nhị phân (có thể được coi như là một ma trận nhị phân – mỗi phần tử của ma trận là một bit) được chia ra thành các khối m x n bit, mỗi khối giấu được một bit thông tin bằng cách thay đổi nhiều nhất là một bit trong khối. Khóa K là một ma trận kích thước m x n.

2.2.2.1 Nội dung thuật toán Wu-Lee

Bước 1 : chia ảnh F thành các ma trận nhỏ F_i kích thước m x n.

Bước 2 : Với mỗi Fi nhận được ở bước 1, kiểm tra điều kiện :

$0 < SUM (F_i \wedge K) < SUM (K)$ có đúng hay không?

Nếu đúng thì chuyển sang bước 3 để giấu một giữ liệu vào F_i, ngược lại thì không có dữ liệu nào giấu vào F_i à F_i sẽ được giữ nguyên.

Bước 3 : giấu bit b vào F_i :

If (SUM (Fi ^ K) mod 2 = b) then

Giữ nguyên Fi không đổi ;

Else if (SUM(F_i ^ K) then

Chọn một bit $[F_i]_{j,k}$ bất kì thoa :($[F_i]_{j,k} = 0$ mà $[K]_{j,k} = 1$);

Thay $[F_i]_{j,k} = 1$;

Else if (SUM (F_i ^ K) = SUM (K) – 1) then

Chọn một bit $[F_i]_{j,k}$ bất kì mà $[F_i]_{j,k} = 1$;

Thay $[F_i]_{j,k} = 0$;

Else

Chọn một bit $[F_i]_{j,k}$ bất kì mà $[F_i]_{j,k} = 1$;

Bổ sung $[F_i]_{j,k}$;

Else If

- Sau khi gắn dữ liệu thì F_i được chuyển thành F_i' và giữ được tính chất bất biến sau đây ;

$0 < SUM (F_i' \wedge K) < SUM (K) \rightarrow \quad SUM (F_i' \wedge K) = b \pmod 2$

2.2.2.2 Phân tích và đánh giá thuật toán

- Thuật toán sử dụng K nhằm làm tăng độ mật cho thuật toán giấu tin. Để tìm được ma trận khóa K thì đã biết m, n các thuật toán thám mã phải duyệt $O(2^{mxn})$ trường hợp khác nhau.
- Theo định nghĩa phép toán ⊗, và nội dung thuật toán Wu-Lee sẽ biến đổi F thành F' sao cho SUM(F' ⊗ K) cùng tính chẵn lẻ với b. Do vậy, nếu b không cùng tính chẵn lẻ với SUM(F' ⊗ K) thì thuật toán sẽ thực hiện đảo giá trị của phần tử $F_{i,j}$ ứng với $K_{i,j} = 1$ để đạt được bất biến. Như vậy, khóa K được xem như một mặt nạ, tạo ra khung nhìn cho thuật toán.
- Điều kiện 0 < SUM(F' ⊗ K) < SUM(K) quy định, nếu mọi vị trí (i,j) của F tại các vị trí $K_{i,j} = 1$ mà $F_{i,j}$ đều bằng = 0 hoặc đều bằng 1 thì không nên giấu tin vì nếu thực hiện giấu dễ bị lộ khóa K.
- Ưu điểm của thuật toán này là tương đối đơn giản. Nhược điểm của thuật toán này là tỉ lệ giấu tin thấp vì mỗi khối chỉ giấu được một bit thông tin, và độ an toàn chưa cao, nếu đối phương đã biết ảnh giấu tin sử dụng thuật toán WL thì chỉ cần xác định được m, n và ma trận khóa là sẽ tìm ra tin giấu.

2.2.3 Thuật toán Chen-Pan-Tseng

Trên cơ sở thuật toán của Wu-Lee như đã trình bày, các tác giả Yu Yan Chen, Hsiang Kuang và Yu Chee Tseng đã phát triển một kỹ thuật giấu tin mới, thuật toán giấu tin CPT [4]. Kỹ thuật này sử dụng một ma trận khóa K và một ma trận trọng số W trong quá trình giấu tin và tách thông tin.

- Quá trình biến đổi khối ảnh F thành F' kích thước mxn để giấu r bit thông tin b = $b_1b_2\ldots b_r$ được thực hiện sao cho :

$$SUM((F' \oplus K) \otimes W = b \bmod (2^r)) \qquad (2.1)$$

- Công thức (1) được sử dụng để tách chuỗi bit b = $b_1b_2\ldots$br từ khối ảnh F'.

2.2.3.1 Tóm tắt nội dung thuật toán CPT

- Dữ liệu vào :

+) Xét trên một ma trận nhị phân $F = (F_{ij})_{mxn}$

+) Kết hợp 1 ma trận khóa nhị phân cấp mxn : $K = (F_{ij})_{mxn}$

+) W là ma trận trong số tự nhiên cấp mxn : $W = (W_{ij})_{mxn}$

+) b là dãy r bit cần giấu vào ma trận F_{mxn} : $b = b_1b_2…b_r$

- Ở đây, ta sử dụng b theo hai định nghĩa : dãy bit và số tự nhiên dạng nhị phân

+) Đặt $r = [\log_2(N + 1)]$,mxn = N.

Sao cho :$\{W_{ij}, 1 \le i \le m, 1 \le j \le n\} = \{1, 2, …2^r - 1\}$

- Hay nói cách khác, ma trận trọng số W cần thỏa mãn : mỗi giá trị của tập {1, 2,…2^r - 1} phải xuất hiện trong W ít nhất 1 lần.
- Các ma trận khóa K và ma trận trọng số W kích thước mxn được sử dụng như các thành phần khóa bí mật : người sử dụng K, W trong quá trình giấu và người nhận cần phải có K, W để khôi phục lại tin đã giấu.
- Dữ liệu ra :

+) Ma trận nhị phân F’ đã được mã hóa thông tin dãy r bit b, mà chúng ta có thể lấy lại được thông tin b từ F’.

Thuật toán giấu tin

Bước 1:

Tính $T = F \oplus K$

Và $r = [\log_2(mxn + 1)]$

Bước 2:

Tính $S = \sum\sum T_{ij} \times W_{ij} \bmod 2^r$. (2.2)

Hay S = SUM[T ⊗ W]. Suy ra : $0 \le S \le 2^r - 1$.

Bước 3:

- Ta xem $b = b_1b_2…b_r$ là giá trị dữ liệu cần giấu dưới dạng sơ số 2. Suy ra $0 \le S \le 2^r - 1$. Mục đích của thuật toán này là thay đổi nhiều nhất hai vị trí trong F để được ma trận F’ mà S’ tương ứng tính được theo công thức (2.2) thỏa mãn :

$S' = b \bmod 2^r$. (2.3)

Tính $\alpha = b - S \bmod 2^r$.

Bước 4:

- Ta cần tìm các ô F_{ij} sao cho S tăng đúng một lượng α khi ta đảo giá trị ở ô F_{ij}. Khi đó ta được :

$S' = b = S + \alpha \bmod 2^r$.

- Ta gọi S_α là tập các ô Fij cần đảo sao cho S' = b. S_α thỏa mãn điều kiện đó khi và chỉ khi :

$S_\alpha = \{F_{ij} (T_{ij} = 0, W_{ij} = \alpha \bmod 2^r)$ or $(T_{ij} = 1, W_{ij} = 2^r - \alpha \bmod 2^r)\}$ (2.4)

- Tính S_α theo công thức (2.4).

Bước 5:

- Xảy ra một trong 3 trường hợp sau :

Nếu S = b (hay α = 0) thì dĩ nhiên ta không cần thay đổi ma trận F.

Nếu $\alpha \neq 0$ và $S_\alpha \neq \varnothing$ ta chỉ cần đảo một ô bất kì F_{ij} thuộc S_α . Thuật toán dừng. Nếu $\alpha \neq 0$ và $S_\alpha = \varnothing$ ta chuyển bước 6.

Bước 6:

- Ta tìm số nguyên h > 1 và nhỏ nhất sao cho $S_{h\alpha} \neq \varnothing$ và $S_{\alpha - h\alpha} \neq \varnothing$.Sự tồn tại của h được chứng minh trong [10]. Khi đó ta đảo một ô bất kì thuộc F_{ij} thuộc $S_{h\alpha}$ và một ô bất kì F_{ij} thuộc $S_{\alpha - h\alpha}$.

2.2.3.2 Phân tích và đánh giá thuật toán

- Thuật toán có thể giấu được r bit vào trong một khối mxn với điều kiện là $2^r <$ mxn
- Và chỉ cần thay đổi nhiều nhất là 2 bit lên một khối. Như vậy, thuật toán này đã có cải tiến rất lớn so với những thuật toán khác chỉ giấu được một bit vào mỗi khối.
- Độ an toàn của thuật toán cũng rất cao thông qua hai ma trận dùng làm khóa để giải tin đó là ma trận trọng số và ma trận khóa. Như vậy độ bảo mật của thuật toán là :

$$C_{mn}^{2^{r-1}} * (2^r - 1)! * (2^r - 1)^{mn-(2^r-1)})$$

- Thuật toán Chen-Pan-Tseng sử dụng một ma trận trọng số nhằm giấu được một dãy nhiều bit vào trong mỗi khối, và ma trận trọng số này cũng chính là một thành phần bí mật cùng với ma trận khóa, do vậy độ an toàn của thuật toán Chen-Pan-Tseng sẽ cao hơn thuật toán Wu-Lee.
- Thuật toán này đương nhiên có thể áp dụng cho ảnh màu và ảnh đa cấp xám. Ta cũng sẽ sử dụng kỹ thuật chọn ra bit quan trọng nhất của mỗi điểm ảnh để xây dựng ma trận hai chiều các bit 0,1 như trong thuật toán với ảnh đen trắng.
- Nếu áp dụng tốt thuật toán này cho ảnh màu thì có thể nói thuật toán đã đạt yêu cầu cơ bản của một ứng dụng giấu tin mật đó là đảm bảo tính ẩn của thông tin giấu, số lượng thông tin giấu cao.

2. 3. Kĩ thuật giấu tin trên ảnh biên

2. 3. 1. Ý tưởng của kĩ thuật

Thuật toán giấu tin được Hongxia Wang, Gouxi Chen, Meng Zhang đề xuất vào tháng 5 năm 2013 [5]. Mục đích của thuật toán là để cải tiến độ bền vững của thuật toán giấu tin trong hình ảnh nhị phân bằng phương pháp kết hợp phép biến đổi hình thái học (tách biên) và thuật toán F5. Đầu tiên ảnh được sử dụng phép biến đổi hình thái học co giãn và tách cạnh để được ảnh biên, sau đó ảnh này được giấu tin bằng thuật toán F5.

2.3.2. Một số khái niệm

2.32.1. Giãn nở ảnh

Phép toán hình thái học được đề xuất của Tiến sĩ J. Serra là giáo viên Mather Wing vào năm 1964. Hình thái toán học dựa trên lý thuyết toán học chặt chẽ và hình học, tập trung vào các hình học và mối quan hệ của hình ảnh. Một số hoạt động của hình thái toán học :

Giãn nở ảnh: ký hiệu phép giãn nở là $\oplus$, X được giãn ra bởi B là $X \oplus B$, nó được định nghĩa như sau:

$$X \oplus B = \{x|[(\hat{B})_x \cap A] \neq \Phi\} \quad (2.5)$$

Trong công thức (2.5): B là các bản đồ của B, nó được định nghĩa là:

$$(\hat{B}) = \{x|x = -b, b \in B\} \quad (2.6)$$

$(\hat{B})_x$ có nghĩa là thay đổi B theo x bit, nó được định nghĩa là :

$$(M)_x = \{y|y = a + x, a \in M\} \quad (2.7)$$

Trong công thức (2.7): quá trình của sự giãn nở của B bởi X là: Thay đổi các điểm ảnh trung tâm của B lần đầu tiên, sau đó thay đổi các giá trị của B cho x, giao của X và B không phải là tập rỗng. Nói cách khác, tổng hợp các giá trị của B với X là, một tập hợp các điểm ảnh trung tâm của B, khi có ít nhất một yếu tố khác không giao nhau giữa của B và X. Do đó , công thức (2.5) có thể được viết như sau:

$$X \oplus B = \{x|[(\hat{B})_x \cap X \subseteq X]\} \quad (2.8)$$

Công thức (2.8) có thể giúp chúng ta hiểu các hoạt động giãn nở của khái niệm về chập. Nếu như B mẫu của chập, có nghĩa là sự giãn nở làm các tách xạ của B về các điểm ảnh trung tâm, và sau đó di chuyển bản đồ liên tục trên X.

- Erode – phép toán của xói mòn là Θ, X bị xói mòn bởi B là X Θ B, nó được định nghĩa là:

$$X \Theta B = \{x| (\hat{B})_x \subseteq X \} \quad (2.9)$$

Công thức (2.9) giải thích rằng kết quả của B xói mòn X là tổng hợp của tất cả các x, trong đó B là dịch x vẫn còn trong X. Nói cách khác, tổng hợp mà B xói mòn X là tổng hợp của vị trí ban đầu của B khi B là hoàn toàn nằm trong X.

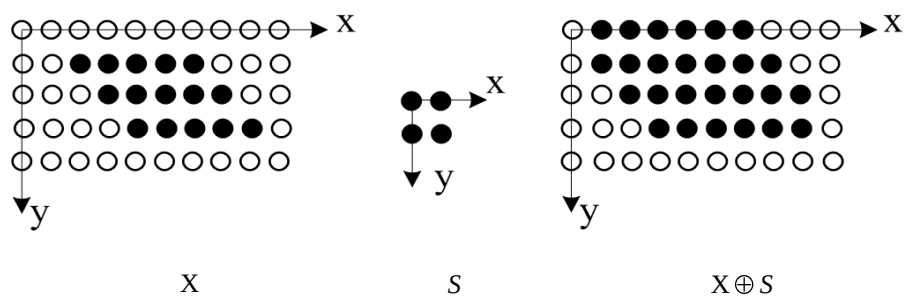
2.3.2.2. Lựa chọn yếu tố cấu trúc

Việc điều chỉnh hình thái toán học cho ảnh được dựa trên khái niệm điền đầy dựa trên các yếu tố cấu trúc, lựa chọn yếu tố cấu trúc và các thông tin của ảnh có mối quan hệ chặt chẽ, chúng ta có thể hoàn thành phân tích hình ảnh khác nhau thông qua xây dựng khác nhau cơ cấu hoàn chỉnh, và có được kết quả thực nghiệm khác nhau.

Dịch các yếu tố cấu trúc S cho x được S_x, nếu S_x và X giao nhau là không có sản phẩm nào, chúng ta ghi lại điểm x, tổng hợp đó được điều chỉnh bởi x đáp ứng các điều kiện trên được gọi là kết quả là S giãn X. Công thức là:

$$X \oplus S = \{x \; S_x \cup x \neq \varnothing\} \quad (2.10)$$

Hình 2.1 là ví dụ minh họa cho ảnh giãn từ ảnh ban đầu X, ma trận cấu trúc S.



Hình 2.1. Ví dụ giãn ảnh

Phương pháp giãn ảnh là so sánh điểm ban đầu của S và điểm ban đầu của X từng người một, nếu một điểm S nằm trong phạm vi của X, thì các điểm tương ứng với điểm ban đầu của S là ảnh kết quả bên phải hình 2.1 là ảnh kết quả giãn. Nó cho thấy, nó có chứa tất cả các phạm vi của X, như X được giãn ra một vòng. Và nếu ma trận cấu trúc khác nhau kết quả của sự giãn nở cũng khác nhau.

2.3.2.3.Tìm biên của ảnh

Chúng ta có thể nhìn thấy từ mô tả ở trên, nếu ảnh được giãn bằng một ma trận yếu tố cấu trúc, nội dung ảnh sẽ giãn ra một vòng. Sau đó, nếu chúng ta để cho ảnh giãn trừ đi ảnh ban đầu, ta sẽ nhận được biên của ảnh trong ảnh nhị phân biên của ảnh xuất hiện như hình thức đột biến của giá trị màu xám. Khi yếu tố cấu trúc có cùng một giá trị (giá trị cùng một màu xám), vì sự khác biệt về giá trị là lớn, giá trị của hình ảnh đầu ra được thay đổi là thấp hơn so với hình ảnh ban đầu. Vì vậy, nếu chúng ta để cho ảnh giãn trừ đi các ảnh ban đầu, có thể nhận được biên của ảnh.

Biểu thức tách biên được biểu diễn qua phép toán hình thái học như sau:

$$(X \oplus B) - X = X \cap (X \oplus B)^C \qquad (2.11)$$

Ở đây chọn ma trận cấu trúc kích cỡ 3 × 3 yếu tố cấu trúc cho các thử nghiệm.

Hình 2.2 là kết quả của ảnh nhị phân được mô phỏng trong hình thái toán học. Trong đó, ảnh ban đầu Hình 2.2 (a) là ảnh thuộc sở hữu của cả người gửi và người nhận, được sử dụng để so sánh ảnh người nhận được giãn với yếu tố cấu trúc kích cỡ 3 × 3, chúng ta được ảnh biên hình 2.2 (c) ảnh này được sử dụng để giấu tin mật.

(a) Hình ảnh nhị phân ban đầu (b)Hình ảnh được giãn (c) Hình ảnh cạnh

Hình 2.2. Tách cạnh trên ảnh nhị phân

2.3.2.4. Khôi phục ảnh bằng phân mảnh ảnh và định danh điều chỉnh

Sau khi tách cạnh của ảnh ban đầu, chúng ta nên điều chỉnh nó thông qua phương pháp phân vùng ảnh và định danh. Chia ảnh biên hình 2.2 (c) thành các mô đun ảnh kích cỡ 3×3. $F_1, F_2, ..., F_{2t}$ tổng cộng 2^t.

Cho F_i (i = 1,3,5 ... 2t-1) là mô-đun định dạng, thì F_j là tổng hợp của module mà các thông tin có thể được nhúng.

Trong F_i thì các mô-đun có điểm ảnh trung bình từ 0,3 đến 0,7 kí hiệu F_u ($F_u \subset F_i$) thì các mô đun tương ứng $F_u + 1$ ($F_{u+1} \subset F_j$) là mô-đun hình ảnh có thể được nhúng vào các thông tin. Ghép các mô-đun có thể nhúng vào ảnh T, T là hình ảnh thực sự có thể được nhúng thông tin.

2.3.3. Thuật toán giấu tin F5

Thuật toán F5 được đề xuất bởi nhà khoa học người Đức Pfitzmann và Westfeld vào năm 2001 [6]. Thuật toán này nhúng thông điệp vào LSB của các hệ số DCT theo bước đi giả ngẫu nhiên thông qua tất cả các hệ số DCT của ảnh cover trong đó nó bỏ qua các hệ số DC và các hệ số bằng 0. Nếu LSB của hệ số DCT không phù hợp với bit thông điệp, giá trị tuyệt đối của hệ số giảm đi 1. Nếu phép trừ dẫn đến 0 thì bit thông điệp phải nhúng vào hệ số tiếp theo, bởi vì ở phía người nhận, thông điệp chỉ được lấy ra ở các hệ số DCT khác 0. Đặc biệt F5 sử dụng ma trận mã hoá (matric encoding) để giảm thiểu số thay đổi cần thiết khi giấu thông điệp. Với cách thức này nó có thể giảm thiểu tối đa được khoảng trên 50% thay đổi trên ảnh so với Jsteg. Theo miêu tả của thuật toán F5 phiên bản 11, chương trình yêu cầu các thông tin vào gồm có

- Yếu tố chất lượng Q của ảnh stego

- Ảnh vào (dạng TIFF, BMP, JPEG, hoặc GIF)
- Tên tệp đầu ra
- Tệp chứa thông điệp bí mật
- Mật khẩu người dùng được sử dụng cho bộ tạo giả ngẫu nhiên PRNG
- Chú thích chèn vào phần header của ảnh (thường là độ dài thông điệp)

Matrix encoding có 3 tham số (c, n, k) với c là số thay đổi trên một nhóm gồm n hệ số DCT, k là số bit được nhúng. Trong [3] tác giả sử dụng một matrix encoding đơn giản (1, 2^k-1, k), sử dụng một hàm băm để đưa ra k bit khi áp dụng cho 2^k –1 hệ số.

Ví dụ, nếu chúng ta muốn nhúng 2 bit x_1, x_2 (k=2) vào nhóm 3 hệ số (2^2-1=3) a_1, a_2, a_3 sẽ chỉ thay đổi một vị trí. Chúng ta có thể bắt gặp 4 trường hợp xẩy ra sau:

$x_1=a_1 \oplus a_3$, $x_2= a_2 \oplus a_3 \Rightarrow$ không thay đổi gì

$x_1 \neq a_1 \oplus a_3$, $x_2= a_2 \oplus a_3 \Rightarrow$ thay đổi a_1

$x_1=a_1 \oplus a_3$, $x_2 \neq a_2 \oplus a_3 \Rightarrow$ thay đổi a_2

$x_1 \neq a_1 \oplus a_3$, $x_2 \neq a_2 \oplus a_3 \Rightarrow$ thay đổi a_3

Từ đó ta có thể lấy ra được x_1, x_2 dựa vào các trường hợp trên.

Trường hợp tổng quát, chúng ta có một từ mã a với n vị trí bit có thể thay đổi cho k bit thông điệp mật x. Đặt f là một hàm băm dùng để lấy ra k bit từ một từ mã. Matrix encoding có thể giúp chúng ta tìm ra một thay đổi phù hợp a thành a’ đổi với mọi a và x với x=f(a’) sao cho khoảng cách Hamming

$$d(a, a') <= d_{max}$$

vì matrix encoding gồm ba phần (d_{max}, n,k) cho nên một từ mã với n vị trí sẽ thay đổi không quá d_{max} vị trí để nhúng k bit. F5 thực hiện maxtrix encoding với d_{max} =1. Đối với (1, n, k) từ mã có độ dài là n=2^k – 1. Khi đó bỏ qua điểm hội tụ (tức hệ số DCT bằng 0) chúng ta sẽ nhận được mật độ thay đổi là

$$D(k) = \frac{1}{n+1} = \frac{1}{2^k}$$

Và tỉ lệ nhúng

$$R(k)=\frac{k}{n}=\frac{1}{n}ld(n+1)=\frac{k}{2^k-1}$$

Sử dụng mật độ thay đổi và tỉ lệ nhúng chúng ta có thể định nghĩa hiệu suất nhúng W(k). Nó có thể chỉ ra giá trị bit trung bình chúng ta có thể nhúng trên sự thay đổi đó

$$W(k)=\frac{R(k)}{D(k)}=\frac{2^k}{2^k-1}.k$$

Hiệu suất nhúng của (1, n, k) luôn luôn lớn hơn k. Bảng 2.1 chỉ ra tỉ lệ nhúng giảm trong khi hiệu suất nhúng tăng. Tuy nhiên chúng ta có thể đạt được hiệu suất cao chỉ với thông điệp rất nhỏ.

Bảng 2.1. Mối liên hệ giữa mật độ thay đổi và tỉ lệ nhúng.

k	N	Mật độ thay đổi (D(k))	Tỉ lệ nhúng (R(k))	Hiệu suất nhúng W(k)
1	1	50.00%	100.00%	2
2	3	25.00%	66.67%	2.67
3	7	12.50%	42.86%	3.43
4	15	6.25%	26.67%	4.27
5	31	3.12%	16.13%	5.16
6	63	1.56%	9.52%	6.09
7	127	0.78%	5.51%	7.06
8	255	0.39%	3.14%	8.03
9	511	0.20%	1.76%	9.02

Bảng 2.2 đưa ra sự phụ thuộc giữa bit thông điệp x_i và vị trí bit được thay đổi a'_j. Chúng ta chia phần phụ thuộc với mã nhị phân của j tới cột a'_j vậy nên chúng ta có thể tìm ra hàm băm rất nhanh.

Khi đó $f(a)=\oplus_{i=1}^{n} a_i.i$

Bảng 2.2. Sự phụ thuộc (×) giữa bit thông điệp x_i và các bit từ mã a'_j.

f(a')	a'_1	a'_2	a'_3
x_1	×		×
x_2		×	×

f(a')	a'_1	a'_2	a'_3	a'_4	a'_5	a'_6	a'_7
x_1	×		×		×		×
x_2		×	×			×	×
x_3				×	×	×	×

Chúng ta có thể tìm ra vi trí bit $s = x \oplus f(a)$

Khi đó từ mã được thay đổi trong $a' = \begin{cases} a \text{ nếu } s = 0 (\Leftrightarrow x = f(a)) \\ (a_1, a_2, ..., \overline{a_s}, ..., a_n) \text{ ngược lại} \end{cases}$

Chúng ta có thể tìm ra một tham số k tốt nhất cho mọi thông điệp để nhúng vào mọi vật mang cung cấp đủ khả năng nhúng thông điệp sao cho thông điệp vừa đủ trong vật mang. Ví dụ, nếu chúng ta muốn nhúng một thông điệp 1000 bit vào một vật mang có khả năng nhúng 50000 bit thì tỉ lệ nhúng cần thiết là R=1000:50000=2%. Giá trị này nằm giữa R(k=8) và R(k=9) trong bảng 2.1. Chúng ta chọn k=8 và có thể nhúng 50000:255=196 từ mã với độ dài n=255, hay matrix encoding là (1,255,8) có thể nhúng 196.8=1568 bit. Nếu chúng ta chọn k = 9 chúng ta không thể nhúng được thông điệp.

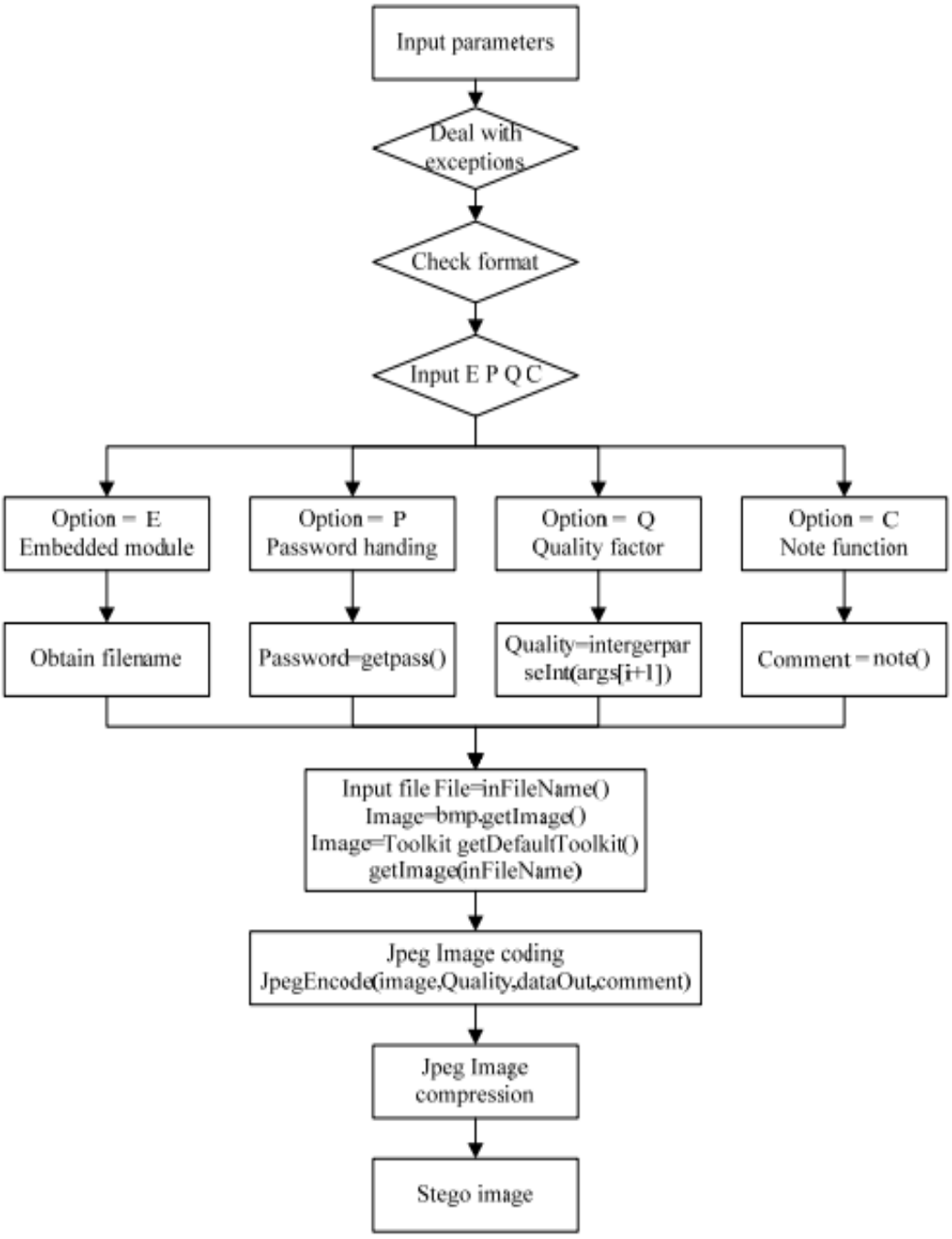
Thuật toán F5 bao gồm các bước sau:

1. Lấy phần thể hiện RGB của ảnh đầu vào (lấy dữ liệu của ảnh).
2. Biến đổi miền dữ liệu của ảnh sang miền tần số DCT sau đó lượng tử hoá các hệ số DCT theo Q ta được các hệ số DCT đã lượng tử.
3. Tính khả năng có thể nhúng khi không sử dụng matrix encoding $C = h_{DCT} - h_{DCT}/64 - h(0) - h(1) + 0.49h$, trong đó h_{DCT} là tông số hệ số DCT, $h(0)$ là số hệ số AC DCT bằng 0, $h(1)$ là số hệ số AC DCT có trị tuyệt đối bằng 1, $h_{DCT}/64$ là số hệ số DC, $-h(1)+0.49h(1) = -0.51h$ là ước lượng mức độ hao hụt.
4. Mật khẩu người dùng được sử dụng để tạo ra bộ khởi tạo giả ngẫu nhiên PRNG cái mà quyết định nhúng các bít thông điệp vào các vị trí ngẫu nhiên. PRNG cũng thường được sử dụng để phát sinh một dòng bít giả ngẫu nhiên

bằng phép XOR với thông điệp tạo ra nó một dòng bít ngẫu nhiên. Trong quá trình nhúng, hệ số DC và các hệ số = 0 thường được bỏ qua.

5. Thông điệp được chia thành các đoạn gồm k bít, mỗi đoạn nhúng vào một nhóm hệ số DCT 2^k–1 theo bước đi giả ngẫu nhiên. Nếu giá trị băm của nhóm không phù hợp với các bít thông điệp, thì giá trị tuyệt đối của một trong những hệ số trong nhóm bị giảm đi 1 cho phù hợp. Nếu hệ số trở thành 0 (hệ số này được gọi là điểm hội tụ - shrinkage), và khi đó k bít thông điệp này sẽ được nhúng trong nhóm hệ số DCT tiếp theo (lưu ý LSB(d) = d mod 2 với $d > 0$, và LSB(d) =1–d mod 2 với $d < 0$).

6. Nếu độ dài thông điệp phù hợp với khả năng có thể giấu trong ảnh thì quá trình giấu thành công, ngược lại sẽ thông báo lỗi và cho biết độ dài lớn nhất của ảnh có thể giấu để điều chỉnh thông điệp giấu hoặc thay đổi ảnh dùng để giấu thông điệp.



Hinh 2.3. Sơ đồ thuật toán F5 [5].

2.3.4. Thuật toán giấu tin và tách tin trên biên bằng F5

2.3.4.1. Thuật toán giấu tin

Thuật toán giấu tin trên ảnh biên như sau :

Đầu vào : ảnh nhị phân X, thông điệp giấu M.

Đầu ra : ảnh đã giấu tin S.

Các bước thực hiện :

Bước 1 : Từ ảnh nhị phân X thực hiện giãn ảnh theo công thức (2.1) với phần tử cấu trúc B được ảnh đã giãn I.

Bước 2 : Từ ảnh I tìm biên theo công thức (2.7) được ảnh E .

Bước 3 : Thực hiện giấu tin trên ảnh E bằng thuật toán F5 được ảnh đã giấu tin Y.

Bước 4 : Thực hiện khôi phục lại ảnh nhị phân bằng công thức :

$$S = I – Y$$

Bước 5 : Lưu trữ ảnh kết quả S

Ví dụ minh họa

Sử dụng ảnh nhị phân baboon.jpg, kích cỡ 512x512 pixel (hình 2.4 a). Thực hiện giãn ảnh (hình 2.4 b) và tìm biên (hình 2.4 c). Thực hiện giấu tin bằng thuật toán F5 với nội dung trong hình 2.4. d), ta được ảnh biên đã giấu tin 2.4 e). Khôi phục lại ảnh nhị phân ban đầu ta được ảnh nhị phân đã giấu tin trên vùng biên 2.4 f).

a)

b)

c)

d) e) f)

Hình 2.4. Quá trình giấu tin: a) ảnh nhị phân ban đầu b) ảnh đã giãn c) ảnh biên d) thông điệp e)ảnh biên đã giấu tin, f) ảnh nhị phân đã giấu tin.

2.3.4.2. Thuật toán tách tin

Thuật toán tách tin trên ảnh biên như sau :

Đầu vào : ảnh nhị phân X đã giấu tin

Đầu ra : thông điệp đã giấu

Các bước thực hiện :

Bước 1 : Từ ảnh nhị phân X thực hiện giãn ảnh theo công thức (2.1) với phần tử cấu trúc B được ảnh đã giãn I.

Bước 2 : Từ ảnh I tìm biên theo công thức (2.7) được ảnh E .

Bước 3 : Thực hiện tách tin trên ảnh E bằng thuật toán tách tin F5 được thông điệp đã giấu M

Ví dụ minh họa

Sử dụng ảnh nhị phân baboon.jpg, kích cỡ 512x512 pixel (hình 2.5 a). Thực hiện giãn ảnh (hình 2.5 b) và tìm biên (hình 2.4 c). Thực hiện tách tin bằng thuật toán F5 t được thông điệp đã giấu M (hình 2.5 e)

a) b) c) d)

Hình 2.5. Quá trình tách tin: a) ảnh nhị phân ban đầu b)ảnh đã giãn c) ảnh biên d) thông điệp.

Chương 3. KẾT QUẢ THỰC NGHIỆM

3.1. Môi trường thử nghiệm

-Ngôn ngữ cài đặt: Ngôn ngữ lập trình Matlab phiên bản 7.7

-Môi trường soạn thảo: Matlab phiên bản 7.7

-Môi trường chạy chương trình: Môi trường giao diện Matlab phiên bản 7.7

-Cấu hình tối thiểu để cài đặt Matlap:

+Intel hoặc AMD x86 processor supporting SSE2

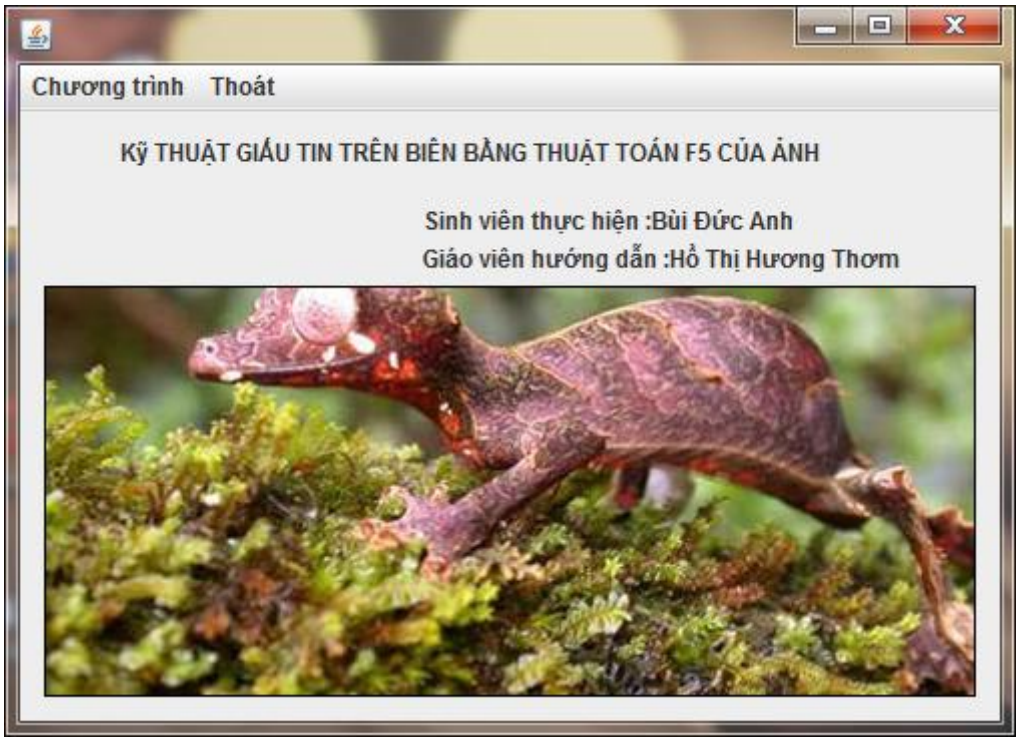
+Dung lượng ổ cứng từ 16GB tới 32GB

+Bộ nhớ RAM tối thiểu 1GB

3.2. Giao diện chương trình

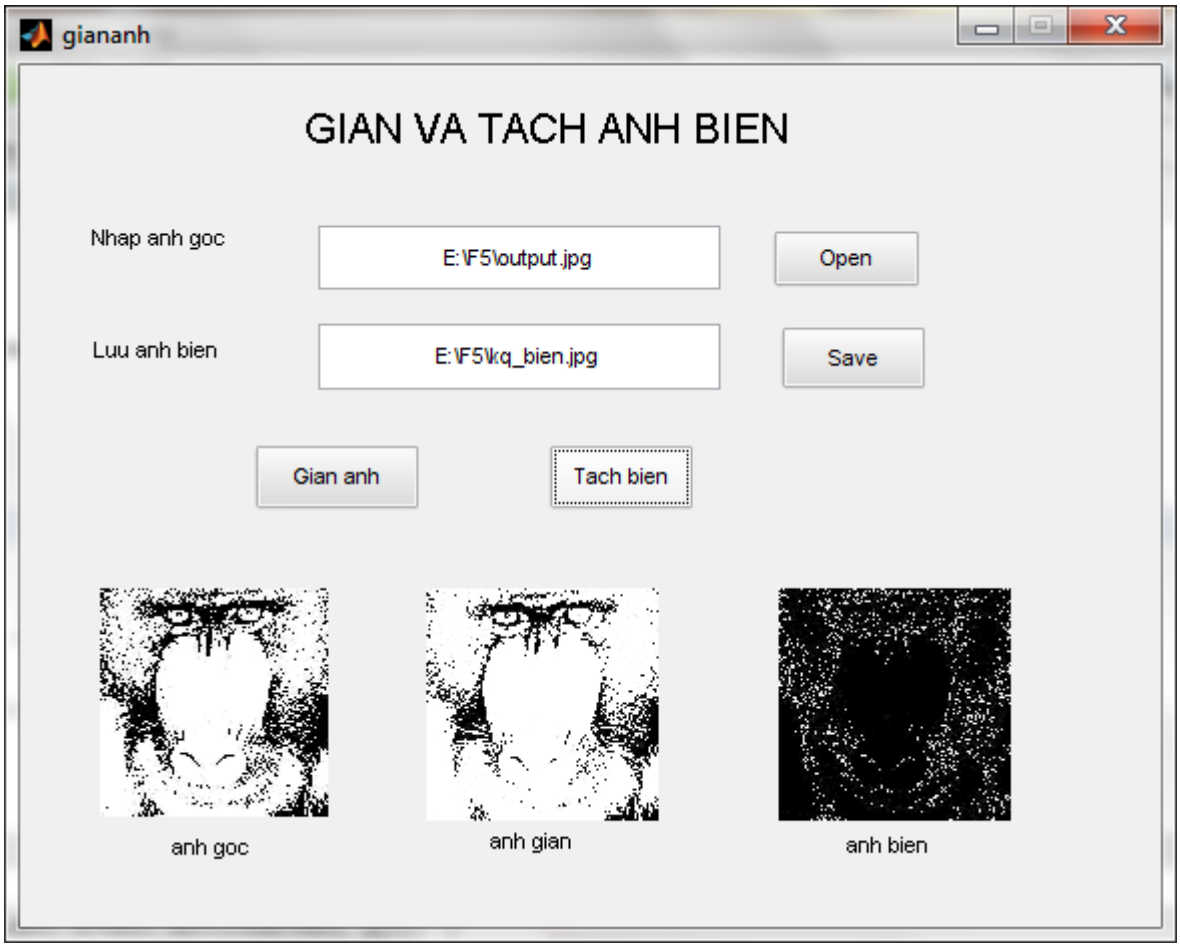
3.2.1 Giao diện chương trình chính

Sau đây là một số giao diện chương trình chính. Trong đó chương trình F5 được tải về từ [6], ban đầu chương trình được xây dựng từ Java chạy trên môi trường DOS, sau đó sinh viên đã phát triển giao diện thân thiện hơn.



Hình 3.1. giao diện chính chương trình.

Đây là giáo diện chính chương trình, từ đấy ta có thể gọi các giao diện khác thông qua các menu.



Hình 3.2. Giao diện chức năng tách và giãn ảnh biên.

3.2.2 Giao diện chương trình giấu tin

Chương trình xây dựng khi thực hiện theo dòng lệnh đánh vào từ Dos có dạng như sau:

java Embed [tuy chon] <ten anh Cover> [ten anh ket qua]

Trong đó:

+ Tuỳ chọn gồm có

-e <file to embed> : cho biet tệp chứa thông điệp cần giấu tin (nếu không có thành phần này sẽ là nén ảnh Jpeg thông thường

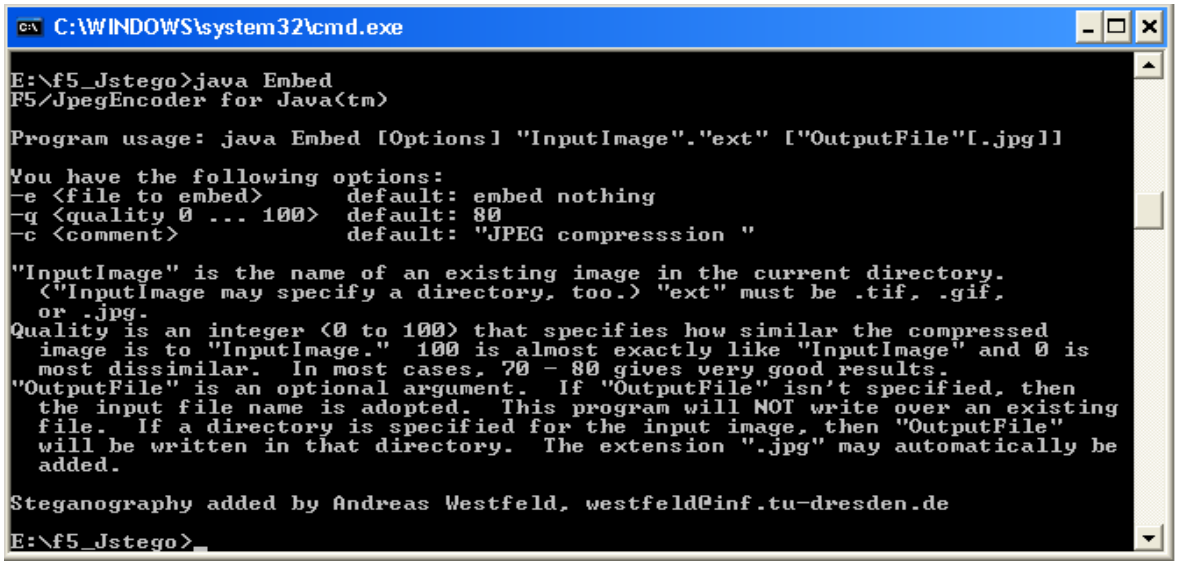
- q <quality 0…100> : cho biết tỉ lệ nén theo Q (mặc định Q=80)

- c <comment>: điền thêm chú thích vào phần đầu tệp Jpeg

+ <InputImage> là tên ảnh cần nén hoặc dùng để chứa thông điệp (ảnh 24 bit màu gồm các dạng Gif, Tif, Bmp hoặc Jpeg)

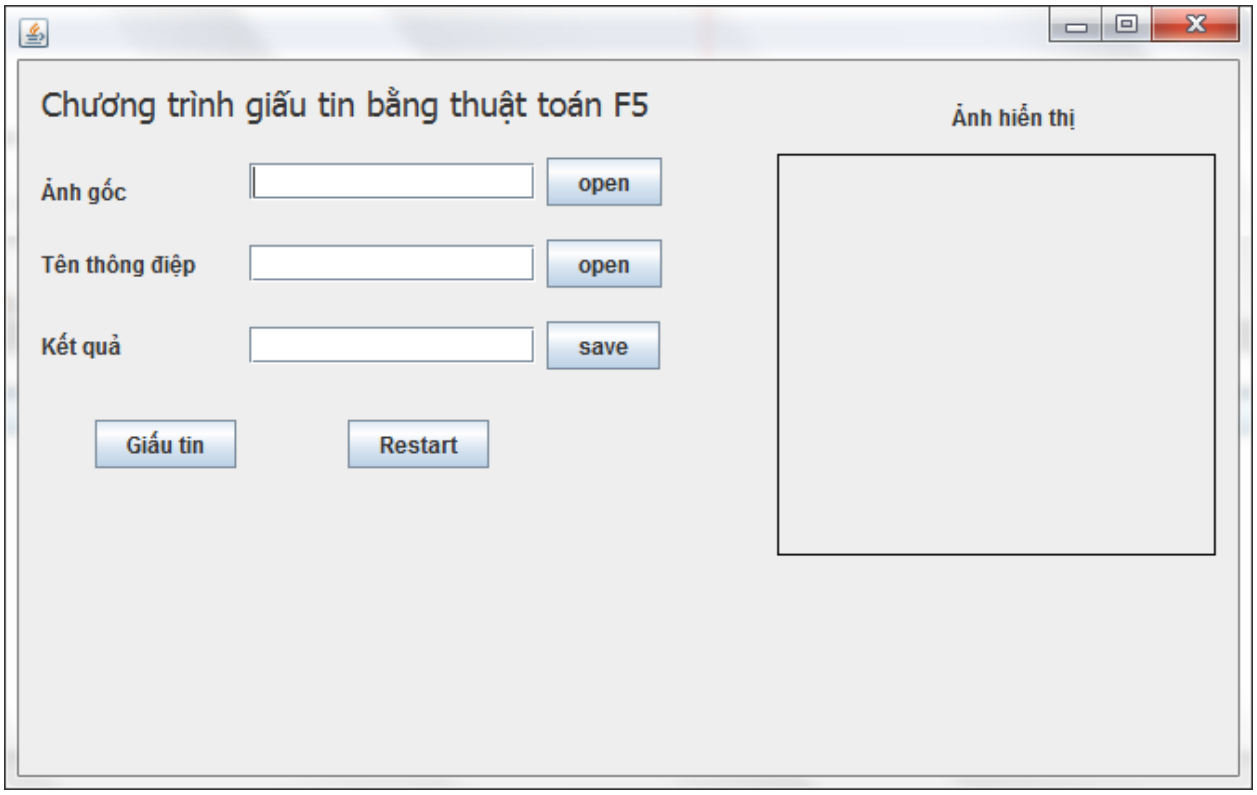
+ [OutputImage] : cho biết tên ảnh nén Jpeg nếu không có thành phần này thì tên ảnh kết quả sẽ giống như tên ảnh cover chỉ khác phần mở rộng (mặc định là .jpg).

Nếu đánh cú pháp ***java Embed*** thì trên màn hình sẽ hiện ra chú thích câu lệnh đầy đủ của chương trình. Chi tiết theo hình vẽ 3.3.



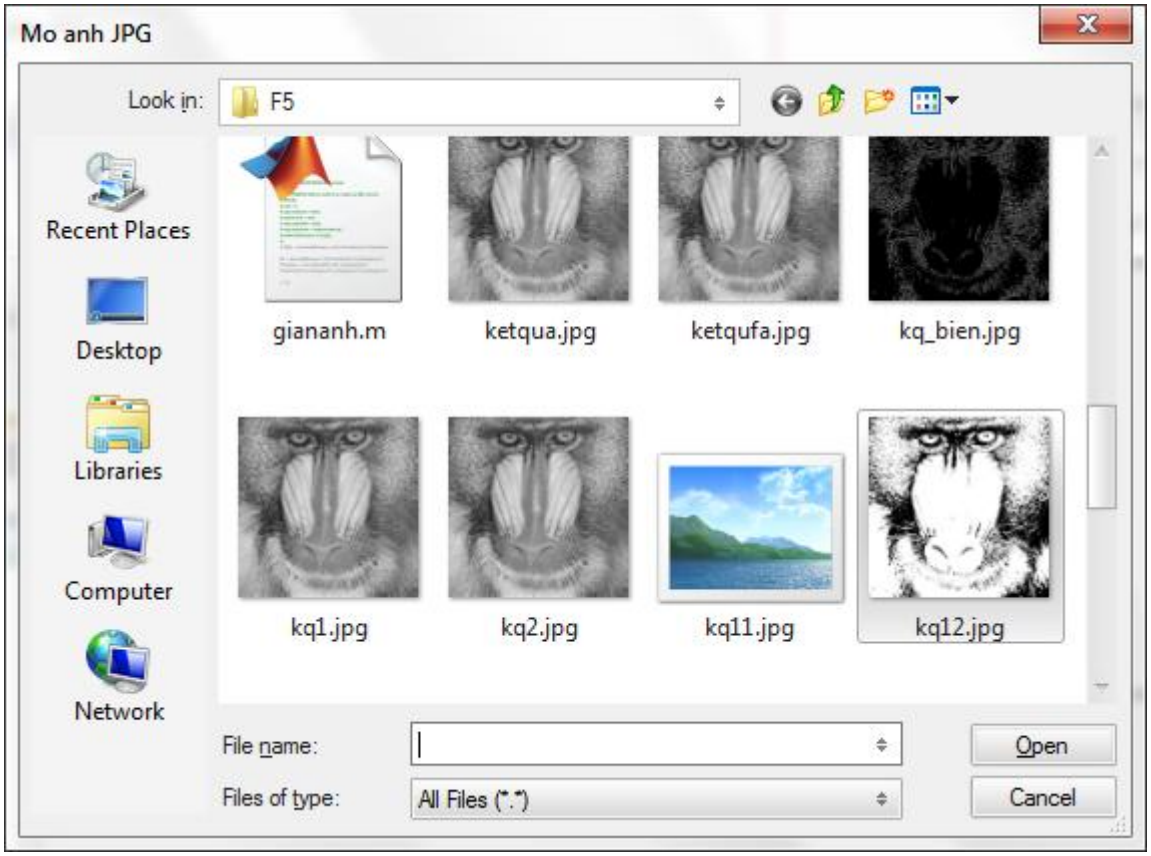
Hình 3.3. Giao diện chương trình nhúng Embed

Sau khi xây dựng giao diện (với chất lượng mặc định Q=80) ta được:



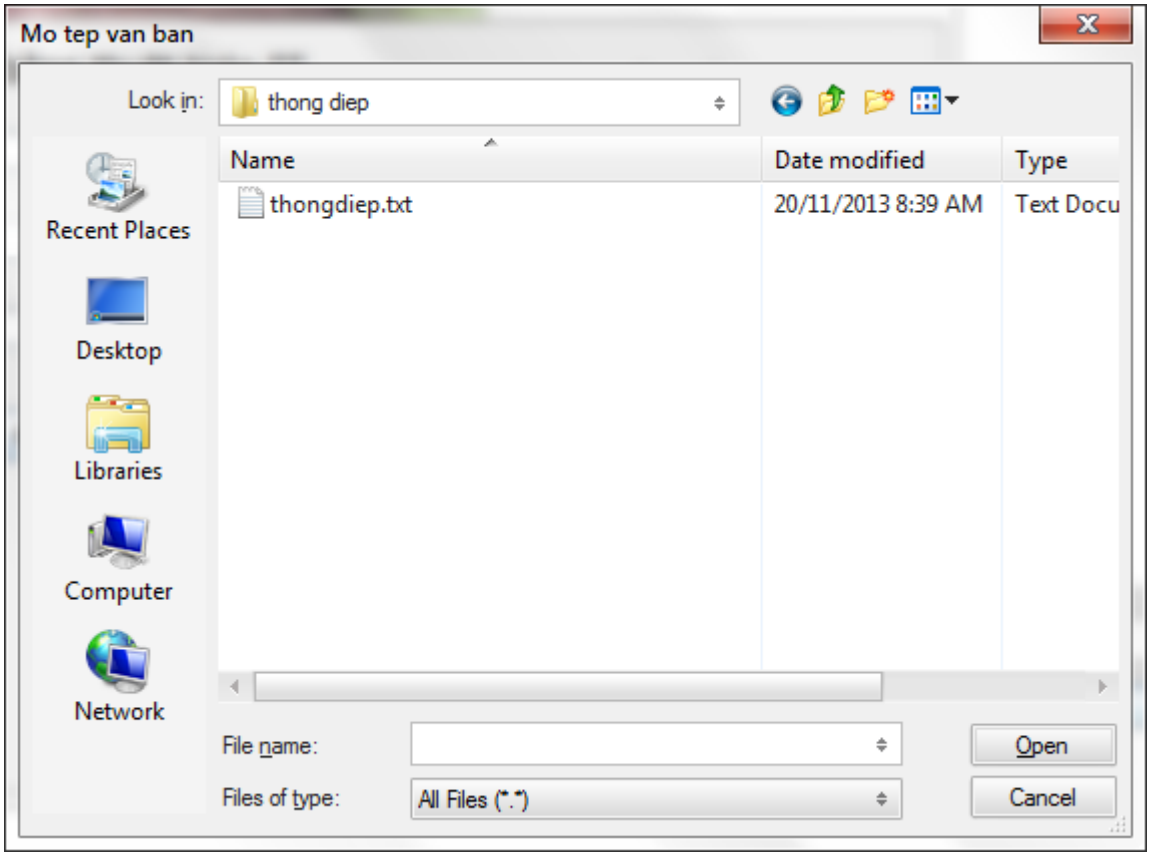
Hình 3.4. Giao diện chức năng giấu tin.

Từ giao diện chính của chương trình chúng ta chọn ảnh cần giấu tin bằng cách nhấn vào button "Open".Khi đó chương trình sẽ mở ra hộp thoại duyệt ảnh.



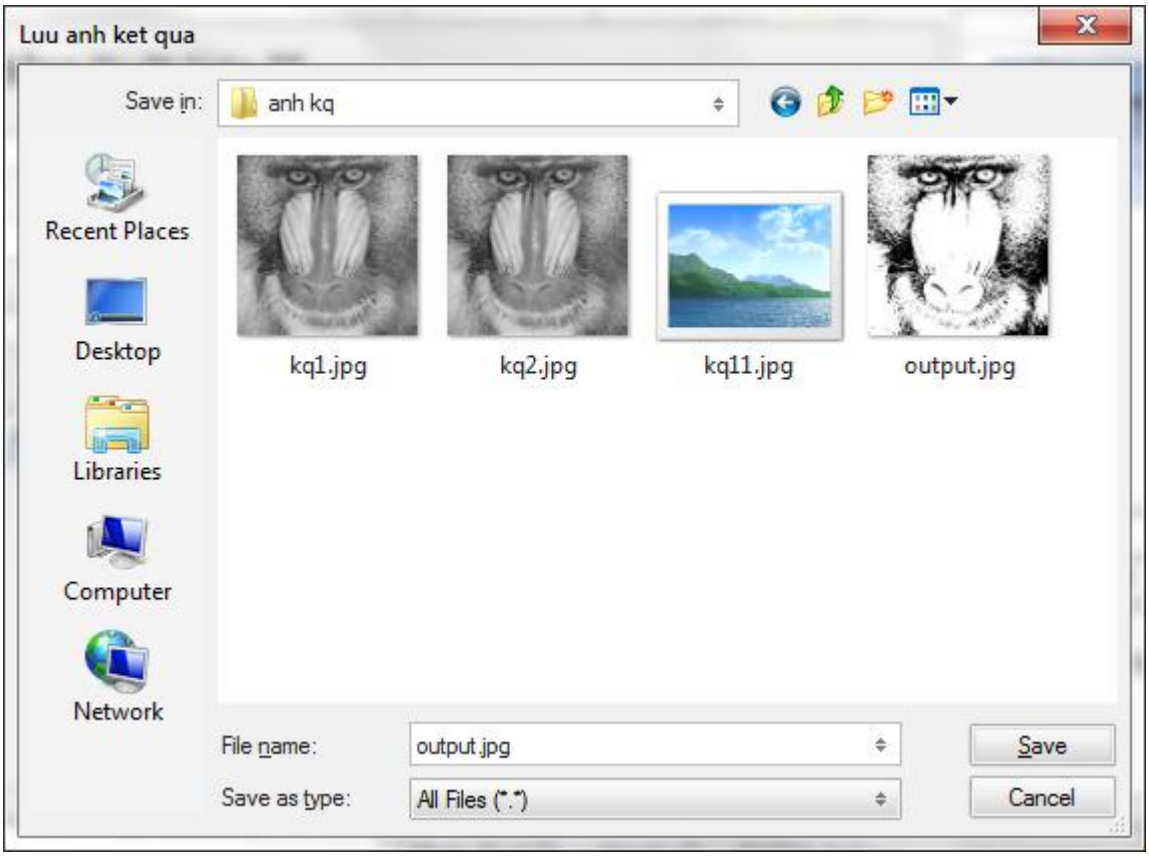
Hình 3.5. Hộp thoại chọn ảnh nhị phân cần giấu tin.

Chúng ta sẽ chọn ảnh nhị phân bất kì để thực hiện giấu tin vào ảnh đó. Sau khi chọn ảnh nhị phân xong, ta nhập thông diệp vào từ bàn phím hoặc lấy thông điệp từ tệp *.txt bất kì để giấu tin.



Hình 3.6. Hộp thoại chọn tệp thông điệp.

Chúng ta cần chọn nơi sẽ lưu thông điệp sau khi đã giấu tin vào bằng cách chọn “Save as” từ giao diện



Hình 3.7. Hộp thoại cho biết tên ảnh sau khi đã giấu tin.

Hình 3.8. Hộp thoại cho biết tên ảnh sau khi đã giấu tin.

Sau khi đã lựa chọn xong đầu vào và đầu ra cho chương trình, chúng ta chọn nút "thực hiện giấu tin". Chương trình sẽ thực hiện và đưa ra kết quả ảnh đã giấu tin ngay trên giao diện của chương trình

3.2.3 Giao diện chương trình tách tin

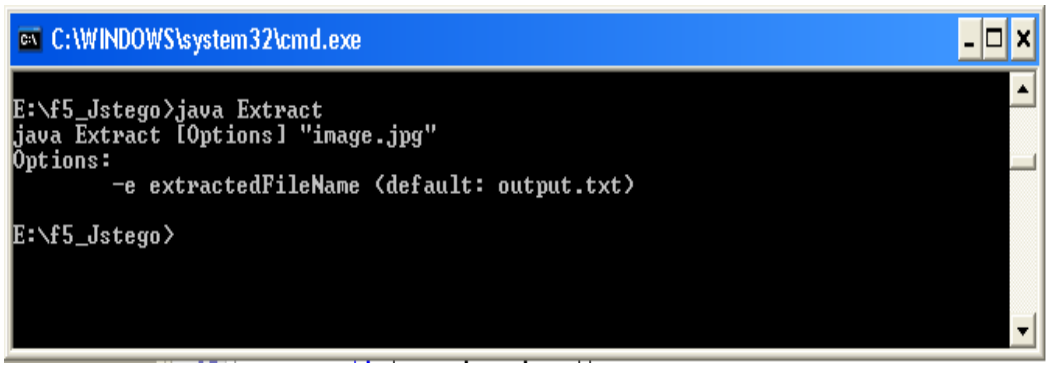
Chương trình tách thông điệp Extract viết bằng Java chạy trên Dos được thực hiện theo cú pháp như sau:

java Extract [tuy chon] <ten anh stego.jpg>

Trong đó:

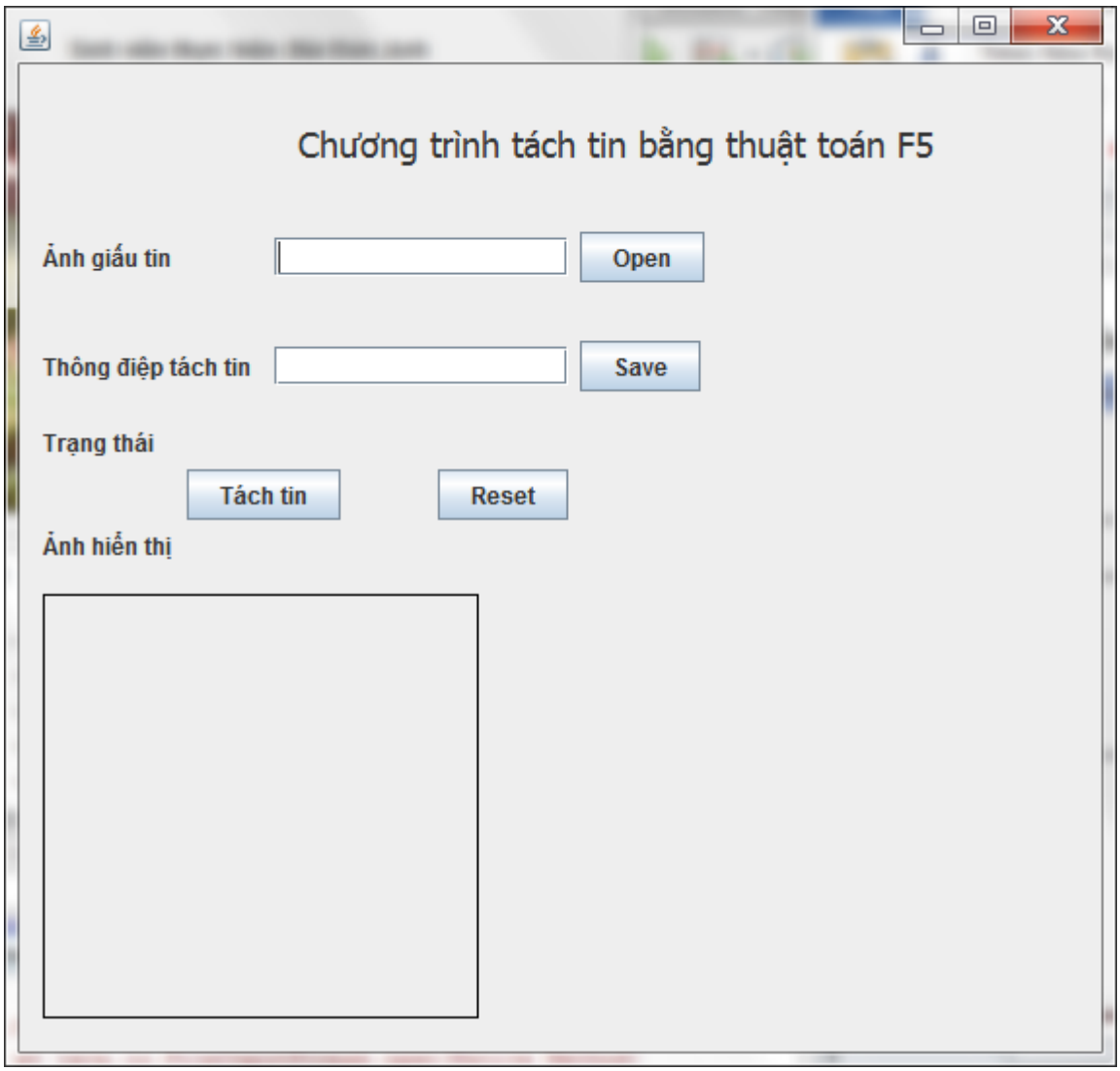
Tuỳ chọn là –e <tep ket qua> : cho biết tên tệp sẽ chứa nội dung thông điệp tách ramặc định là tệp output.txt nếu người sử dụng không gõ chi tiết.

Nếu người sử dụng gõ *java Extract* chương trình sẽ hiện ra chú thích đầy đủ cú pháp câu lệnh. Chi tiết theo hình 5.8.



Hình 3.9. Giao diện tách thông điệp Extract.

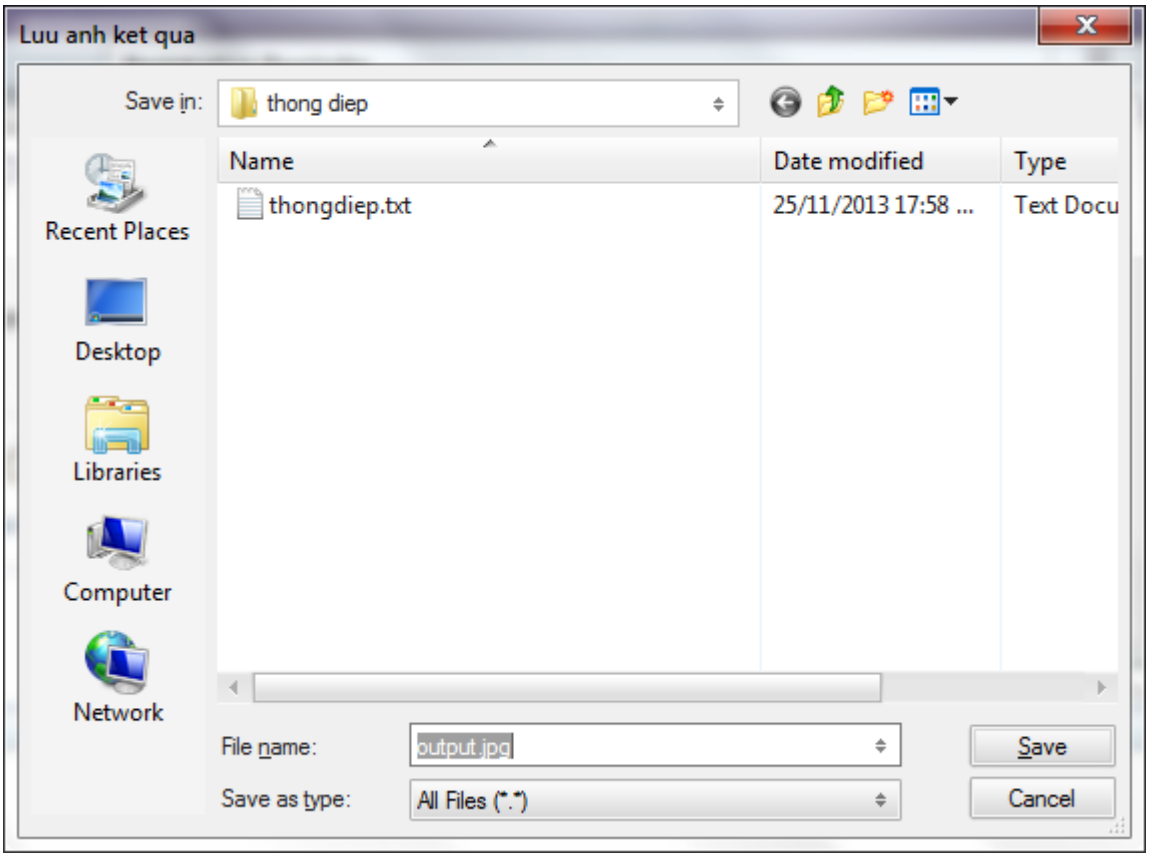
Sau khi xây dựng giao diện ta được:



Hình 3.10. Giao diện chức năng tách tin.

Bước đầu cần cho biết ảnh mang tin bằng cách chọn "open" để lấy giá trị đầu vào của ảnh. Sau đó ta tiến hành tách tin bằng cách nhấn button "Thực hiện tách tin".

Sau khi đã tách tin xong, chúng ta tiến hành lưu lại thông điệp bằng cách chọn button "Lưu thông diệp"



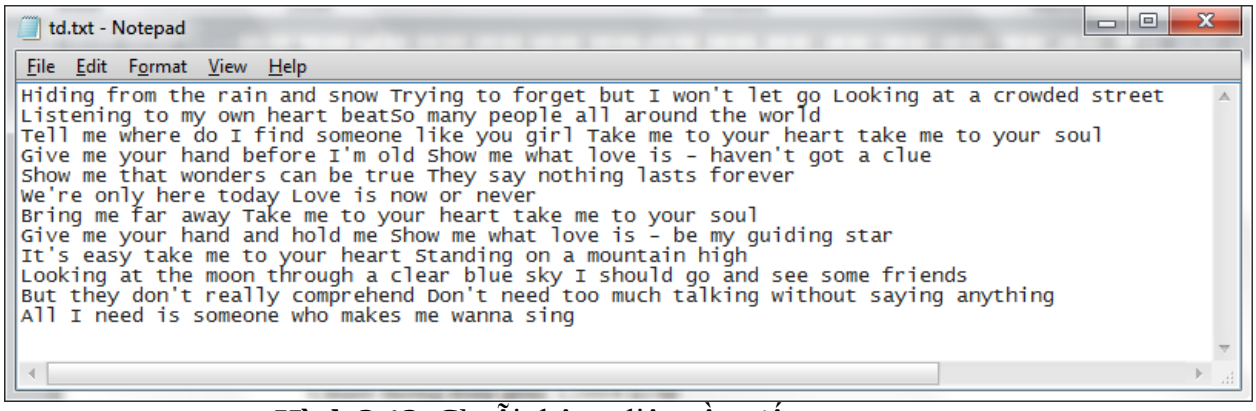
Hình 3.11. Hộp thoại chọn lưu thông điệp.

Chúng ta gõ tên của tệp thông điệp cần lưu lại và chọn "Save" để tiến hành lưu lại trên máy tính.

3.3. Kết quả thực nghiệm và nhận xét

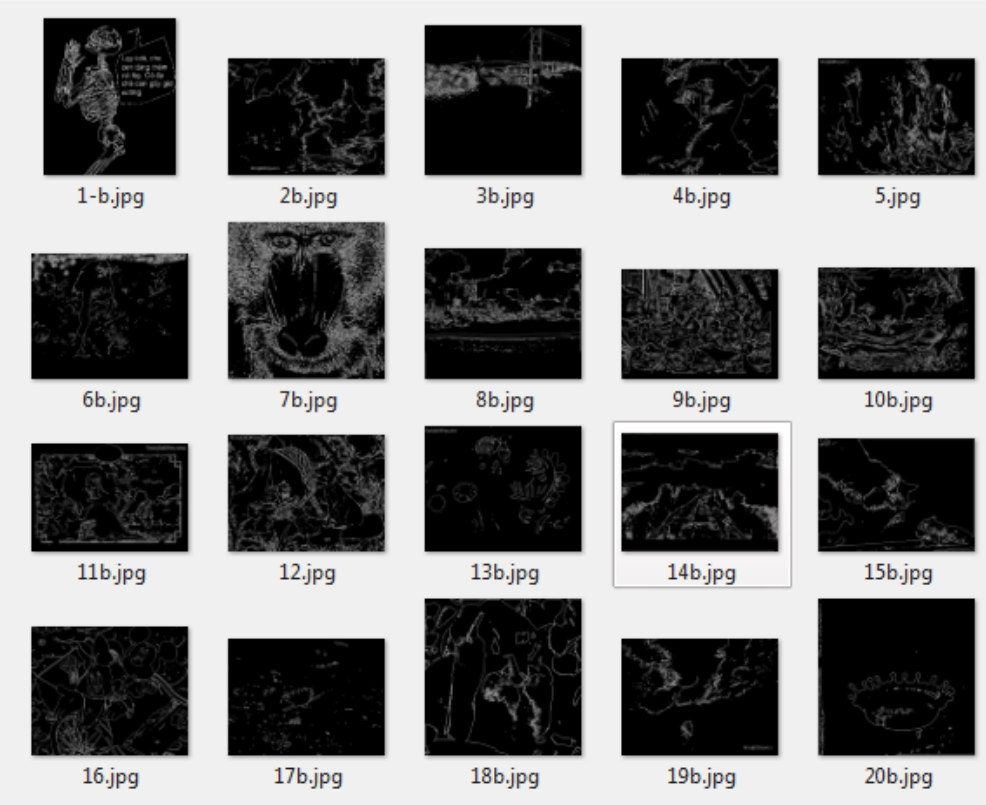
3.3.1. Kết quả thực nghiệm

Thực nghiệm này sẽ đưa ra khả năng giấu tin khi sử dụng kỹ thuật giấu tin trên biên của ảnh nhị phân.



Hình 3.12. Chuỗi thông điệp cần giấu.

Hình 3.13. Tập ảnh trước khi giấu tin.



Hình 3.14. Tập ảnh biên.

Hình 3.15. Tập ảnh thử nghiệm.

Đánh giá PSNR đơn vị đo là dB

Bảng 3.1. Kết quả đánh giá PSNR của ảnh sau khi giấu tin.

Tên ảnh (kích cỡ ảnh)	**Đánh giá PSNR (dB)**
1.bmp (293x349)	22.7441
8.bmp (700x525)	23.4497
10.bmp (647x619)	23.4916
11.bmp (700x525)	18.5448
12.bmp (700x525)	19.4293
13.bmp (777x624)	23.4113
14.bmp (512x512)	22.2758
15.bmp (746x619)	22.4526
17.bmp (732x510)	18.3116
18.bmp (731x515)	22.4037
19.bmp (700x479)	21.4827
20.bmp (700x525)	20.4215
22.bmp (749x603)	24.457
24.bmp (762x581)	23.501

34.bmp (700x501)	23.4926
38.bmp(788x647)	23.4529
43.bmp (640x480)	23.4688
69.bmp (700x525)	23.4567
a.bmp (512x512)	23.4318
b.bmp (512x512)	23.5084
Giá trị trung bình	21.4094

3.3.2. Nhận xét

Với kết quả thử nghiệm thu được, nếu quan sát bằng mắt thường thì có thể phân biệt được đâu là ảnh đã giấu tin và chưa giấu tin. Giá trị PSNR trung bình đạt được là bình thường khi giấu lượng bit thông điệp tương đối lớn.

Kết quả thử nghiệm trong bảng 3.1 cho thấy khả năng giấu tin của mỗi ảnh khác nhau là khác nhau. Những ảnh cùng kích cỡ khả năng giấu của những ảnh đó nằm trong một khoảng giá trị và xấp xỉ bằng nhau. Điều đó chứng tỏ khả năng giấu phụ thuộc vào giá trị điểm ảnh của ảnh.

Qua thử nghiệm em nhận thấy kỹ thuật giấu tin trên biên của ảnh nhị phân có những ưu nhược điểm sau:

- Ưu điểm:

  + Khả năng bảo mật cao.

  + Không ảnh hưởng nhiều đến nội dung của ảnh.

- Nhược điểm:

  + Độ dài thông điệp phải phù hợp với khả năng có thể giấu trong ảnh thì quá trình giấu mới thành công.

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 việc che giấu thông tin trong ảnh số cũng như trình bày kỹ thuật giấu tin trên biên của ảnh nhị phân.

Với kỹ thuật giấu tin trên biên của ảnh nhị phân thì tính vô hình của thông tin sau khi giấu được đảm bảo. Về mặt lý thuyết thì sau khi đã có lượng thông tin được giấu vào trong ảng gốc, nó sẽ để lại dù nhiều, dù ít những dấu vết khác với ảnh gốc ban đầu. Tuy nhiên sau khi thực hiện kỹ thuật giấu tin, quan sát bằng mắt thường thì khó có thể phân biệt đâu là ảnh gốc đâu là ảnh mang tin. Dùng phương pháp đánh giá PSNR để đánh giá chất lượng ảnh trước và sau khi giấu tin kết quả PSNR đạt được là có thể chấp nhận được, điều đó cho thấy sự biến dạng của ảnh hầu như không có. Như vậy kỹ thuật giấu tin đã cho những kết quả rất triển vọng.

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 trên biên của ảnh nhị phân.

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 của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

[1]. Bùi Thế Hồng (2005); "*Về một cải tiến đối với lược đồ giấu dữ liệu an toàn và vô hình trong các bức ảnh hai màu*", Tạp chí Tin học và điều khiển học, tập 21, số 4-2005, pp281-292.

[2]. M. Y. Wu and J. H. Lee (1988); "*A Novel Data Embedding Method for Two-Color Facsimile Images*". In Proceedings of International Symposium on Multimedia Information Processing, Chung-Li, Taiwan, R. O. C, December 1998

[3]. Phan Trung Huy, Vu Phuong Bac, Nguyen Manh Thang, Truong DucManh, Vu Tien Duc, Nguyen Tuan Nam, "*A New CPT Extension Scheme for High Data Embedding Ratio in Binary Image*s", the Proceedings of the 1st KSE. Inter. Conf. Hanoi 10/2009. 61-66. IEEE.CS.

[4].Yu Yuan Chen, Hsiang Kuang Pan and Yu Chee Tseng (2000); "*A Secure Data Hiding Scheme for Two-Color Images*", IEEE Symp. on Computer and Communication.

[5]. Hongxia Wang, Gouxi Chen Meng Zhang, *Edge Steganography for Binary Image*, TELKOMNIKA, Vol. 11, No. 5, May 2013, pp. 2822 ~ 2829.

[6]. Andreas Westfeld*: High Capacity Despite Better Steganalysis (F5–A Steganographic Algorithm).*In: Moskowitz, I.S. (eds.): Information Hiding. 4th International Workshop. Lecture Notes in Computer Science, Vol.2137. Springer-Verlag, Berlin Heidelberg New York (2001) 289– 302.(https://code.google.com/p/f5-steganography/)