**3. Chương 3: Ứng dụng kỹ thuật DCT được sử dụng trong thủy vân cho ảnh số**

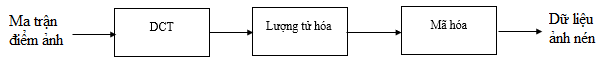
Ở chương 3, em sẽ trình bày chi tiết về kỹ thuật biến đổi cosine rời rạc (DCT) trong thủy vân cho ảnh số. Trước hết, em sẽ giới thiệu về lý thuyết kỹ thuật DCT trong việc nén và giải nén ảnh số. Đây là cơ sở quan trọng cho việc nhúng thủy vân dựa trên kỹ thuật DCT. Tiếp đó, em sẽ trình bày cụ thể về cơ sở lý thuyết toán học của phép biến đổi DCT trong việc thủy vân số kèm theo ví dụ mình họa. Cuối cùng, em sẽ mô phỏng việc thủy vân bằng kỹ thuật DCT bằng ngôn ngữ lập trình Python và phân tích kết quả đạt được.

* 1. **Kỹ thuật DCT trong nén và giải nén ảnh số**

Hiện nay, để đáp ứng với nhu cầu sử dụng dữ liệu rất lớn của con người thì phải cần có các phương pháp để lưu trữ dữ liệu một cách tối ưu. Ví dụ: Khi chúng ta lướt web, trên trang web đó có thể chứa từ vài trăm đế vài nghìn bức ảnh số. Nếu như chúng ta lưu trữ số lượng lớn các bức ảnh thô như vậy thì sẽ phải cần một không gian lưu trữ khổng lồ, điều này là hết sức lãng phí. Thay vì đó, chúng ta có thể sử dụng kỹ thuật nén ảnh để giảm kích cỡ lưu trữ của ảnh. Có 2 loại kỹ thuật chính đó là nén tổn hao (lossy compression) và nén không tổn hao (loseless compression). JPEG là một kỹ thuật nén tổn hao được sử dụng rộng rãi. Nó được sử dụng bởi kỹ thuật DCT trong việc xử lý ảnh. Nói chung, kỹ thuật DCT chia ảnh ra thành nhiều vùng với nhiều mức tần số khác nhau, sau đó trong quá trình xử lý lượng tử hóa, những vùng tần số không quan trọng với mắt người được loại bỏ và chỉ giữ lại những vùng tần số quan trọng với mắt người. Chính vì vậy, khi tiến hành quá trình giải nén ảnh, các những vùng quan trọng của ảnh sẽ được giữ nguyên còn những vùng không quan trọng sẽ có nhiễu.

* + 1. ***Nén ảnh số với DCT***

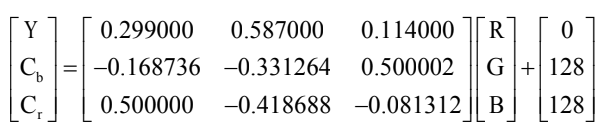
Biến đổi cosine rời rạc (DCT) là một công cụ toán học để xử lý hình ảnh số. Nó cho phép chuyển đổi các điểm ảnh từ miền làm việc không gian sang miền làm việc tần số và ngược lại mà không gây ảnh hưởng đến chất lượng của ảnh. Cơ sở lý thuyết của kỹ thuật nén ảnh số sẽ được trình bày dưới đây.

**

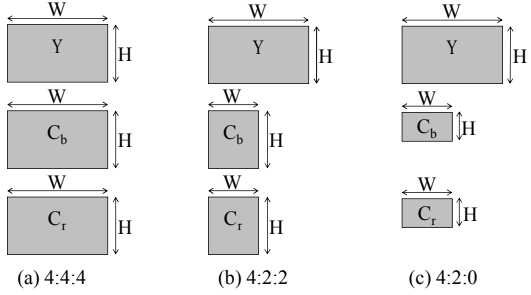
*Hình 3.1: Sơ đồ tổng quan các bước nén ảnh số bằng kỹ thuật DCT*

Các bước thực hiện nén ảnh số với DCT:

* Bước 1: Nếu ảnh là ảnh màu , nó sẽ được chuyển sang không gian màu Trong đó là độ chói của ảnh và là độ sáng của thành phần màu (xanh lam và đỏ) như sau:



Vì mắt người có đặc điểm là ít nhạy cảm với màu sắc hơn nên ở bước này ta có thể giảm dữ liệu ảnh ở vùng mà không làm ảnh hưởng đến chất lượng trực giác của ảnh. Đây gọi là bước giảm mẫu (downsampling):

**

*Hình 3.2: Giảm mẫu trong vùng không gian màu*

Sau đó, ảnh được chia ra thành các khối 8x8 điểm ảnh. Mỗi khối sẽ tương ứng với một ma trận các điểm ảnh với giá trị thuộc khoảng (0,256). Tiếp theo, ảnh sẽ được chuẩn hóa giá trị trong khoảng từ (-128,127) bằng cách trừ đi 128 của giá trị mỗi điểm ảnh

* Bước 2: DCT được thực hiện với từng khối theo thứ tự từ trái qua phải, từ trên xuống dưới theo công thức 3.1. Sau khi thực hiện DCT xong, ta sẽ có 64 sóng cosine cho mỗi khối, điều đó có nghĩa là mỗi điểm ảnh sẽ được thay thế bởi một sóng cosine. Ma trận DCT sẽ biểu diễn ảnh gốc theo cách tách biệt được vùng điểm ảnh quan trọng và ít quan trọng đối với mắt người. Chính vì DCT có tính chất có thể biến đổi ngược nên chúng ta có thể phục hồi lại dữ liệu ở công đoạn giải nén ảnh với nhiễu là rất nhỏ.

Công thức biến đổi DCT và DCT ngược lần lượt là:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
|  |  |  |

Trong đó:

* : ma trận điểm ảnh sau khi thực hiện DCT
* : ma trận điểm ảnh gốc

Để có được dạng ma trận D như ở phương trình 3.1, để đơn giản hóa quá trình DCT chúng ta sử dụng ma trận trực giao T được xác định bởi phương trình sau:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Đến đây, chúng ta sẽ thực hiện DCT bởi công thức nhân ma trận sau:

|  |  |  |
| --- | --- | --- |
|  |  |  |

* Bước 3: Mỗi khối sẽ được lượng tử hóa. Đây là bước mà dữ liệu ảnh sẽ được nén. Sau khi áp dụng DCT, ta lượng tử hóa ma trận DCT với mục đích là để bỏ đi các vùng điểm ảnh có tần số cao bởi vì đây là vùng không quan trọng với hệ thống mắt người. Với việc lượng tử hóa, ta sẽ sử dụng một bảng lượng tử hóa đã được quy định sẵn. Bước này được thực hiện theo công thức sau:

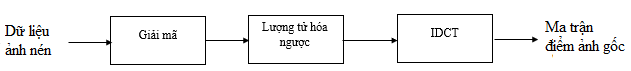
|  |  |  |
| --- | --- | --- |
|  |  |  |

Trong đó:

* Ma trận sau khi lượng tử hóa
* : Ma trận lượng tử hóa tiêu chuẩn
* Bước 5: Ma trận của các khối của điểm ảnh sẽ được mã hóa bằng phương pháp mã hóa Zig-Zag và Huffman. Sau đó dữ liệu nén sẽ được lưu trữ lại sau khi lượng tử hóa
  + 1. ***Giải nén ảnh số dựa trên kỹ thuật DCT***

Đối với giải nén ảnh số, các bước sẽ ngược lại so với nén ảnh số:

* Bước 1: từ chuỗi bit dữ liệu ảnh, ta tiến hành giải mã Huffman và Zig-zag, từ đó khôi phục lại được ma trận lượng tử hóa



*Hình 3.3: Sơ đồ tổng quan các bước giải nén ảnh số bằng kỹ thuật DCT*

* Bước 2: Ta tiến hành đồng thời lượng tử hóa ngược (dequantitization) và chuẩn hóa ngược (denormalization) để khôi phục lại ma trận DCT
* Bước 3: Sau khi có ma trận DCT, thực hiện biến đổi DCT ngược (IDCT) để khôi phục lại ma trận điểm ảnh ban đầu theo công thức 3.2. Tuy nhiên, để đơn giản hóa quá trình IDCT, khi thực hiện biến đổi DCT ngược, ta cần sử dụng ma trận chuyển vị trực giao T’
* Bước 4: Nếu ảnh là ảnh màu, ta tiến hành chuyển đổi ngược ma trận điểm ảnh trong không gian YCbCr sang RGB, ta được ảnh gốc.
  1. ***Kỹ thuật DCT trong việc giấu thủy vân***

Phương pháp biến đổi cosin rời rạc DCT lần đầu tiên được phát minh vào năm 1974. Cho đến ngày nay, kỹ thuật DCT cũng là cơ sở cho nhiều kỹ thuật xử lý ảnh số nói chung. Nổi bật nhất phải kể đến chuẩn nén ảnh JPEG rất nổi tiếng và được sử dụng rộng rãi hiện nay.

Do DCT là cơ sở của việc nén và giải nén ảnh số nên việc sử dụng kỹ thuật này để nhúng thủy vân vào ảnh số sẽ bền vững và chống lại được sự xâm hại thủy vân từ việc nén tổn hao ảnh.

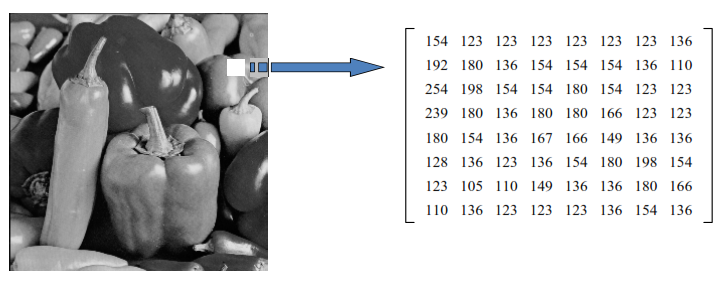
Ở trong phạm vi của đồ án, em xin phép được đi sâu vào trình bày thuật toán thực hiện nhúng và tách thủy vân bao gồm việc tính toán các ma trận hệ số điểm ảnh và bỏ qua các bước tiền xử lý ảnh như chuyển đổi RGB sang không gian YUV và mã hóa dữ liệu ảnh như Mã hóa Zig-Zag và mã hóa Huffman.

* + 1. ***Các bước thực hiện nhúng***

Như đã đã trình bày mô tả thuật toán ở mục 2.3.4, chi tiết về các bước thực hiện thủy vân trên ảnh số sẽ diễn ra như sau:

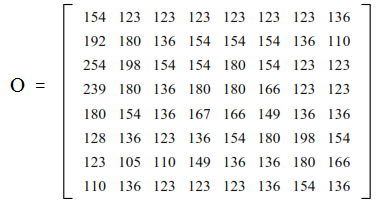
* Bước 1: Chia ảnh gốc ra thành các khối điểm ảnh 8x8

Giả sử ta có ảnh gốc và khối điểm ảnh 8x8 ngẫu nhiên như sau:

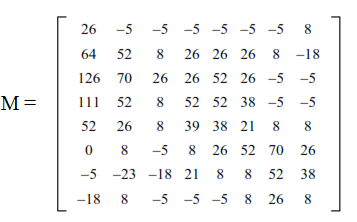


*Hình 3.4: Ma trận điểm ảnh ngẫu nhiên lấy từ một ảnh*

Ta có ma trận gốc O như sau:

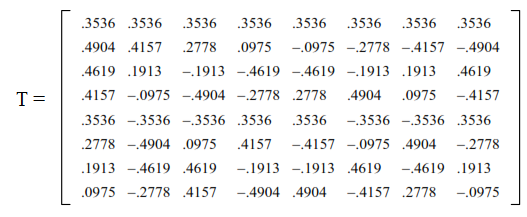


Theo quy tắc DCT thì các giá trị điểm ảnh sẽ nằm trong khoảng (-128,127), vì vậy tất cả các điểm ảnh trong khối sẽ trừ đi 128. Đây được gọi là quá trình chuẩn hóa (normalization) các hệ số của khối điểm ảnh. Ta được ma trận M như sau:



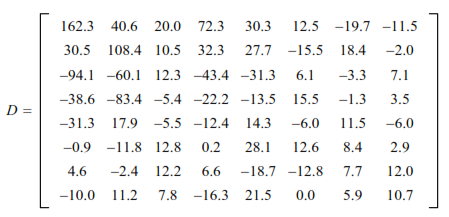
* Bước 2: Thực hiện DCT trên từng khối 8x8

Trước tiên, ta tính ma trận trực giao T theo công thức 3.3. Với ma trận 8x8, ta luôn có ma trận T:

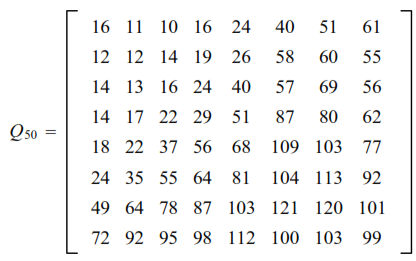


Với hàng đầu tiên (i = 0) ma trận T luôn có giá trị = . Ta tăng dần i và tính các phần tử còn lại của ma trận dựa trên phương trình 3.3. Ma trận T được gọi là ma trận trực giao.

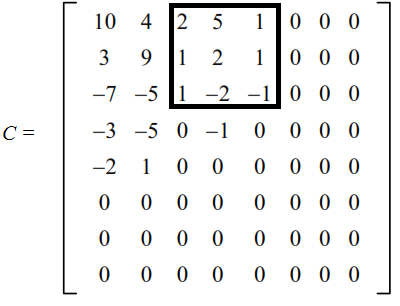
Đến đây, chúng ta sẽ thực hiện DCT bởi công thức 3.4 và được ma trận sau khi đã biến đổi DCT:



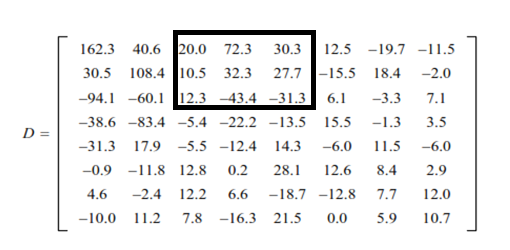
Sau đó, ta tiến hành lượng tử hóa ma trận D theo công thức 3.4 với ma trận lượng tử hóa tiêu chuẩn



Sau khi lượng tử hóa D, ta được ma trận C mới:



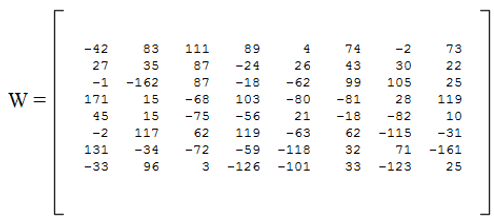
Trên đây là ma trận DCT sau khi lượng tử hóa bao gồm 64 hệ số , với thuộc khoảng (0,7). Hệ số trên cùng tương ứng với tần số thấp của khối ảnh gốc. Càng xa thì hệ số tương ứng càng cao, với tương ứng với tần số cao nhất. Mắt người nhạy cảm nhất với tần số thấp và kém nhạy cảm nhất với vùng tần số cao. Vì vậy những điểm có giá trị 0 có thể bị loại bỏ. Những vùng được đánh dấu là vùng có tần số trung bình. Vì vậy, ta sẽ lựa chọn vùng tương ứng để nhúng thủy vân cho vào ma trận D



Ta thấy rằng các hệ số ở phía góc trái trên cùng của ma trận C sẽ tương ứng với vùng tần số thấp hơn, đây cũng là vùng nhạy cảm với mắt người trong khối ảnh, được sử dụng để tái tạo ảnh trong quá trình giải nén ảnh sau này. Các điểm 0 của ma trận C tương ứng với vùng tần số cao. Đây là vùng không quan trọng với mắt người nên có thể bỏ đi trong quá trình nén tổn hao.

* Bước 3: Nhúng thủy vân dựa trên các vùng hệ số thích hợp

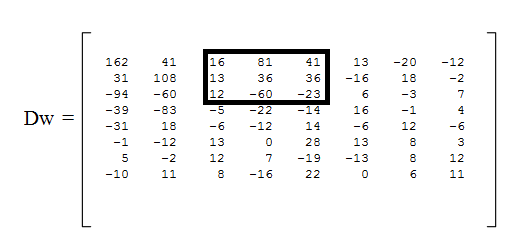
Trước tiên, ta làm tương tự như bước 1 và bước 2 với ảnh thủy vân để ra được ma trận DCT của ảnh thủy vân. Ta có ma trận W như sau:



Sau đó, ta nhúng thủy vân vào ảnh bằng cách hiệu chỉnh hệ số của ma trận D theo công thức:

|  |  |  |
| --- | --- | --- |
|  |  |  |

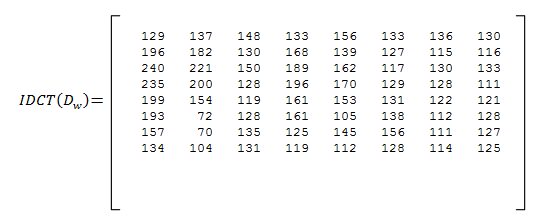
Ở đây ta chọn và , ta được ma trận (làm tròn đến số nguyên gần nhất):



Sau khi kết thúc bước này, ta tiến hành lặp lại từ bước 1 cho đến bước 3 cho tới khi thủy vân được nhúng hoàn toàn vào ảnh gốc

* Bước 4: Biến đổi ngược các khối sử dụng IDCT ta được ma trận ảnh đã được nhúng thủy vân. Sau đó, ta cộng từng phần tử của ma trận với 128 và làm tròn đến số nguyên gần nhất.

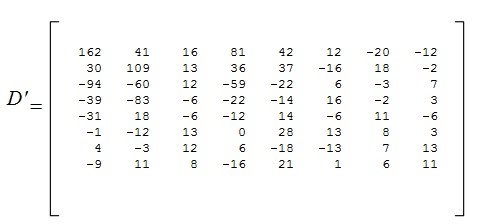
Ta có thể thấy là sau khi biến đổi ngược thì ma trận sau khi tái tạo và có sự thay đổi so với ma trận gốc trong các hệ số của chúng. Lý do là tại các bước IDCT và DCT ta trải qua các bước làm tròn đến số nguyên gần nhất trước khi tính toán.



* + 1. Các bước thực hiện tách thủy vân

Đối với việc tách thủy vân, ta làm ngược lại so với nhúng thủy vân:

* Bước 1: Khi trích xuất ảnh ta thu được ma trận 8x8 điểm ảnh gốc là ma trận
* Bước 2: Thực hiện DCT với ma trận sau khi đã được chuẩn hóa: , ta được kết quả:



* Bước 3: Thực hiện tách hệ số của thủy vân ra khỏi theo công thức:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Ta thu được một phần ma trận là :

Tiếp tục thực hiện lặp lại từ bước 1 tới bước 3 với các khối 8x8 còn lại của ảnh để ra được ma trận hoàn chỉnh

* 1. **Đánh giá chất lượng ảnh sau khi thủy vân**

Việc đánh giá, so sánh một cách chính xác sự sai lệch về chất lượng ảnh gốc S và ảnh sau khi nhúng thông tin thủy vân S’ có thể thực hiện qua việc tính toán giá trị tỷ số tín hiệu đỉnh trên nhiễu PSNR giữa ảnh gốc S và ảnh chứa thủy vân S’ (S và S’ có cùng kích cỡ nhỏ nhất). PSNR được tính theo công thức:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Trong đó:

|  |  |  |
| --- | --- | --- |
|  |  |  |

* : kích cỡ chiều dài và chiều rộng của 2 ảnh

Ở đây, là đại lượng đặc trưng cho sự sai khác giữa ảnh gốc và ảnh sau khi thủy vân. Vi vậy, ta luôn mong muốn giá trị MSE là nhỏ nhất. Khi tức là hai ảnh được so sánh đồng nhất với nhau nên giá trị PSNR không xác định

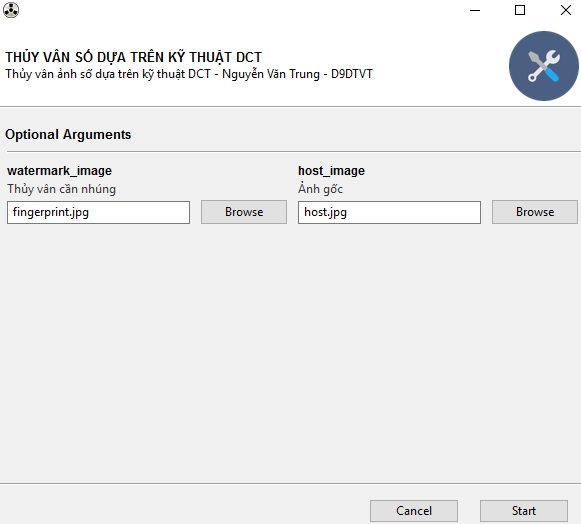
Giá trị thông thường của PSNR sau khi xử lý ảnh thường nằm trong từ 30dB đến 50 dB, Giá trị PSNR càng lớn thể hiện sự sai khác giữa ảnh gốc và ảnh sau khi nhúng thủy vân càng thấp.

* 1. **Mô phỏng kỹ thuật DCT trong thủy vân cho ảnh số sử dụng ngôn ngữ lập trình Python**

Ở phần này, em sẽ mô phỏng kỹ thuật thủy vân cho ảnh số bằng kỹ thuật DCT đã nêu ở trên bằng Python. Python là một ngôn ngữ lập trình bậc cao, nó hỗ trợ rất nhiều thư viện có sẵn dùng để xử lý ảnh như OpenCV, Numpy,… Hơn nữa, Python là ngôn ngữ khá dễ viết, dễ đọc và có giao diện thân thiện với người dùng. Chính vì vậy, em quyết định sử dụng Python để mô phỏng cho sản phẩm thủy vân của mình.

* + 1. Mô phỏng kỹ thuật thủy vân số DCT

Dưới đây là giao diện của chương trình:



Hình 3.5: Giao diện chương trình mô phỏng

Hình 3.6: Ảnh gốc Hình 3.7: Ảnh thủy vân

Trước hết, ta sử dụng ảnh gốc là ‘host.jpg’ và ảnh thủy vân là ‘fingerprint.jpg’ như trên. Sau đó, ta tiến hành nhúng thực hiện nhúng bằng cách nhấn vào nút Start. Ở đây, em thực hiện nhúng ảnh với các mức tỉ số nén khác nhau từ 0% đến 5%. Điều này có nghĩa là chất lượng ảnh sẽ giảm từ 100% xuống 95%. Ta được kết quả như sau

|  |  |  |  |
| --- | --- | --- | --- |
| Chất lượng ảnh(%) | PSNR (dB) | Ảnh gốc | Ảnh sau khi thủy vân |
| 100 | 45.887 |  |  |
| 99 | 43.662 |  |
| 98 | 42.690 |  |
| 97 | 41.712 |  |
| 96 | 40.883 |  |
| 95 | 40.137 |  |

*Bảng 3.1: Kết quả mô phỏng sau khi thủy vân ảnh gốc*

Sau khi tách thủy vân với chất lượng ảnh như trên, ta thu được thủy vân như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Chất lượng ảnh(%) | PSNR(dB) | Thủy vân gốc | Thủy vân sau khi tách |
| 100 | 31.015 |  |  |
| 99 | 30.990 |  |
| 98 | 30.982 |  |
| 97 | 30.979 |  |
| 96 | 30.967 |  |
| 95 | 30.972 |  |

*Bảng 3.2: Kết quả sau khi tách thủy vân*

* + 1. Nhận xét

Về tính ẩn: Đối với kết quả của ảnh thủy vân và ảnh sau khi thủy vân, ta thấy với chất lượng ảnh từ 95% đến 100%, về mặt trực giác gần như ta không nhận thấy được sự khác biệt nào đáng kể giữa các ảnh. Chính vì vậy nên độ ẩn của thủy vân được đảm bảo. Chất lượng ảnh càng tăng thì giá trị PSNR càng tăng.

Về tính bền vững: Đối với thủy vân gốc và thủy vân được khôi phục, ta thấy giá trị PSNR của kỹ thuật DCT khoảng 31 dB với chất lượng ảnh 100%. Điều này có nghĩa là sau khi tách thủy vân phải chịu nhiễu khá lớn. Đối với ảnh đạt 95%, mắt người gần như không thấy được thủy vân hồi phục vì nhiễu lớn. Đây cũng là nhược điểm cần khắc phục của kỹ thuật DCT

Về tính bảo mật: Chỉ có tác giả mới biết được các hệ số nhúng vì vậy nên tính bảo mật của thuật toán được đảm bảo.

**KẾT LUẬN**: Như vậy, ở chương 3, em đã trình bày chi tiết về kỹ thuật DCT sử dụng trong thủy vân cho ảnh số và cơ sở lý thuyết liên quan. Đồng thời, em cũng tiến hành thực hiện chương trình mô phỏng nhúng và tách thủy vân bằng công cụ lập trình Python và đánh giá được kết quả thu được

Tài liệu tham khảo :

[1]: Saraju P.Mohanty, Watermarking of Digital Images, 2004.

[2]: Trần Thị Tú Uyên, Hệ thống thủy vân số và ứng dụng thủy vân số trong việc bảo vệ bản quyền ảnh số, 2017.

[3]: Hebah H.O.Nasereddin, Digital watermarking a technology overview, 2011.

Tài liệu tham khảo:

[1] *Ảnh số* - <http://forum.dohoafx.com/threads/anh-so-la-gi.24401/#.W-4ri-gza00>

[2] *Fundamental of Multimedia*, Ze-Nian Li and Mark S. Drew

[3] *Bài 1- Giới thiệu khái quát về ảnh số* - <http://www.matlabthayhai.info/2015/11/bai-1-gioi-thieu-khai-quat-ve-anh-so.html>

[4] Matlab cơ bản, chuỗi Fourier, biến đổi Fourier- <http://vimach.net/threads/matlab-co-ban-chuoi-fourier-bien-doi-fourier.163/>

[5] *Watermarking in Frequency Domain A Review*, Shahid Bashir Dar, Aasif Bashir Dar, 2014

[6*] Different Watermarking Techniques & its Applications: A Review*, Upasana Yadav,J.P.Sharma, Dinesh.Sharma , Purnima K Sharma , 2014