Implementation of DCT Based Steganography Algorithm to Hide Information in Images

Septrian Napitupulu[1], Ryan Revaldo Siagian[2], Ramli Simanullang[3], Kelvin Daniel Manalu[4], Conglii Fao Silalahi[5]
Fakultas Ilmu Komputer Universitas Katolik Santo Thomas, Jl. Setia Budi No.479F, Tanjung Sari, Kec. Medan Selayang, Kota Medan, Sumatera Utara 20135 , Indonesia

Article Info

Keywords:

Steganography, Discrete Cosine Transform (DCT), Information Hiding, Image Processing, Data Security.

ABSTRACT

Steganography is the art and science of hiding secret messages in digital media, ensuring that only the sender and intended recipient are aware of its existence. One of the widely used techniques in digital image steganography is the Discrete Cosine Transform (DCT) based method, which embeds secret data into the frequency domain of an image. This study explores the application of DCT-based steganography to hide information in an image while maintaining its visual integrity. The research methodology involves transforming the image into DCT coefficients, embedding the secret message into selected high-frequency coefficients, and then applying the Inverse Discrete Cosine Transform (IDCT) to reconstruct the stego image. The quality of the modified image is evaluated using the Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE) metrics. The results show that the proposed method effectively hides information while maintaining image quality, as indicated by the high PSNR value and minimal MSE. This study contributes to the field of secure data communication by presenting a reliable and undetectable approach to information hiding.

Corresponding Author:

Septrian Napitupulu
Fakultas Ilmu Komputer Universitas Katolik Santo Thomas, Jl. Setia Budi No.479F, Tanjung Sari, Kec. Medan Selayang, Kota Medan, Sumatera Utara 20135 , Indonesia
E-mail: korespondensi-penulis@gmail.com

1. INTRODUCTION

Steganography is a technique used to hide information in digital media to ensure secure communication without arousing suspicion.(Based et al., 2024a). Unlike cryptography, which encrypts data and makes it unreadable to unauthorized parties, steganography hides the existence of the message itself. With the rapid advancement of digital communication, the need for secure information exchange has become increasingly important. Various methods have been developed for steganography applications, especially in digital images, including Least Significant Bit (LSB) substitution, Transform Domain techniques, and Adaptive Steganography(Marzuki, 2011).

One of the most widely used techniques for hiding information in images is Discrete Cosine Transform (DCT) based steganography, which works by embedding data into the frequency domain of an image. This method is particularly effective in JPEG images, as the DCT is an essential part of the compression process.(Anilahtirt, 2012). By modifying the high-frequency DCT coefficients, hidden messages can be embedded without significantly changing the visual quality of the image.(As & One, 2021). However, the main challenge in steganography is to maintain a balance between security, capacity, and invisibility, ensuring that the hidden message remains undetected while maintaining the integrity of the image. (Lutfi & Rosihan, 2018).

Several studies have explored various approaches to improve the efficiency and robustness of DCT-based steganography. Westfeld and Pfitzmann (2000) introduced the concept of statistical undetectability in steganographic systems, emphasizing the importance of minimizing distortions that can reveal the presence of hidden information. In addition, research by Fridrich (2009) focused on developing more secure embedding techniques to counter steganalysis attacks. Recent advances in steganography leverage artificial intelligence and deep learning to improve the security of hidden messages.(Based et al., 2024b).

This study aims to implement and analyze DCT-based steganography to hide information in images while maintaining visual quality. This study focuses on evaluating the effectiveness of the method using metrics such as Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE) to assess image distortion. By improving the security and obscurity of steganography techniques, this study contributes to the field of secure data transmission.

2. METHOD

This study applies Discrete Cosine Transform (DCT) based steganography to hide secret information in digital images. This methodology consists of several stages, including data preprocessing, DCT transformation, message embedding, inverse transformation, and quality evaluation.

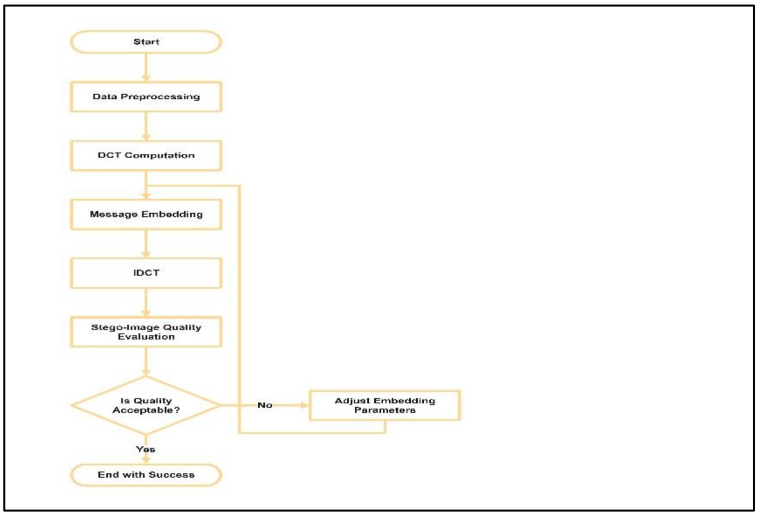

Figure 1. DCT Algorithm Process Flowchart

Data Preprocessing

The study begins by selecting a grayscale image as the masking medium. The image is divided into non-overlapping 8x8 pixel blocks, which are then normalized by subtracting 128 from each pixel value to center the intensity range around zero. This step is necessary to prepare the image for DCT transformation.

$$A(x,y) = P(x,y) - 128$$

Discrete Cosine Transform (DCT) Calculation

Each 8x8 block undergoes a DCT transformation, which converts the spatial domain data into frequency domain coefficients. The transformation is defined as follows:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} A(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

WhereI(xIndonesian:y) represents the pixel intensity, and$\alpha(u)$,$\alpha(v)$ is a normalization factor to ensure energy conservation. The resulting coefficient consists of low-frequency, mid-frequency, and high-frequency components.

Message Pinning

The secret message is first converted into 8-bit binary format based on ASCII values. The embedding process modifies selected high-frequency DCT coefficients to ensure minimal perceptual distortion. The Least Significant Bit (LSB) of these coefficients is replaced with bits from the secret message. The modified DCT coefficients are then stored for reconstruction.

Inverse Discrete Cosine Transform (IDCT)

After embedding, the modified coefficients undergo Inverse Discrete Cosine Transform (IDCT) to reconstruct the stego-image. IDCT is given by:

$$A(x,y) = \sum_{u=0}^{7}\sum_{v=0}^{7} C(u,v)\,\alpha(u)\alpha(v)\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right]$$

Stego Image Quality Evaluation

To measure the effectiveness of the embedding process, stego images are analyzed using two main metrics:

1. Peak Signal to Noise Ratio (PSNR)

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

where the Mean Square Error (MSE) is calculated as:

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(I(i,j) - K(i,j))^2$$

WhereI(iIndonesian:j) is the original image andI(iIndonesian:j) is a stego image.

2. Mean Square Error (MSE)
   This metric measures the mean squared difference between the original image and the stego image, where lower MSE values indicate better invisibility.

3. RESULTS AND DISCUSSION

This section presents the results of the application of DCT-Based Steganography to hide secret messages in digital images. The evaluation includes the steps of image processing, message embedding, extraction, and quality analysis using Peak Signal-to-Noise Ratio (PSNR) and Mean Square Error (MSE). The entire process is implemented using Python.

Manual Implementation of DCT Based Steganography

Discrete Cosine Transform (DCT) based steganography is one of the techniques often used to hide data in digital images. The following is an example of manual calculation of how secret messages can be embedded in the DCT coefficients in an image. Example Implementation in the image:

Figure 1.RGB Image Image 2. Grayscale Image

Figure 3.8x8 Grayscale Image

Initial Pixel Block (8×8)

Suppose we have a grayscale image block of size 8×8 pixels with the following intensity values:

Table 1. Initial Pixel Block (8×8)

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Pixel Block Normalization

Before applying DCT, we need to subtract the average value of 128 from each pixel. Equation (i, j) = P(i, j) - 128

Table 2 Normalized matrix

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-73	-38	-19	-43	-59	-56
-66	-69	-60	-15	-16	-24	-62	-55
-65	-70	-57	-6	-26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-60	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

DCT Calculation

The DCT transform for matrix A(i, j) is calculated using the formula:

$$C(u,v) = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7}A(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

With and as normalization for the coefficients u and v:$\alpha(u)\alpha(v)$

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ u = 0 \\ 1, & if\ u\ \neq 0 \end{cases}$$

The first coefficient C(0.0), which is the DCT coefficient of DC (low frequency), is calculated as follows. Calculation for C(0.0):

$$C(0,0) = \frac{1}{4}\sum_{x=0}^{7}\sum_{y=0}^{7}A(x,y)$$

Substitute the values of A(x, y) into the formula:

$$C(0,0) = \frac{1}{4}(-76 + -73 + -67 + -62 + -58 + -67 + -64 + -55 + \cdots + -34) = -415$$

Calculation for C(1,2):

$$C(1,2) = \frac{1}{4}\alpha(1)\alpha(2)\sum_{x=0}^{7}\sum_{y=0}^{7}A(x,y)\cos\left[\frac{(2x+1)1\pi}{16}\right]\cos\left[\frac{(2y+1)2\pi}{16}\right]$$

We need to calculate the cosine for each value of x and y, then multiply it by the value of A(x, y), and finally multiply the result by . In this case, after numerical calculations, we get:$\alpha(1)\alpha(2)$

$C(1,2) \approx 35$

Calculation for C(2,1):

Using the same formula for the coefficient C(2,1):

$$C(2,1) = \frac{1}{4}\alpha(2)\alpha(1)\sum_{x=0}^{7}\sum_{y=0}^{7} A(x,y)\cos\left[\frac{(2x+1)2\pi}{16}\right]\cos\left[\frac{(2y+1)1\pi}{16}\right]$$

After calculation, we get:

$C(2,1) \approx 28$

The formula for calculating the coefficient applies to all coefficients you want to calculate next.

'HASUGIAN' Message Insertion

The message "HASUGIAN" is converted into binary using the ASCII code for each character. If combined, we will get the binary message: 01001000 01000001 01010011 01010101 01000111 01001001 01000001 01001110.After inserting all the binary message bits into the DCT coefficients, we get the modified DCT coefficients as follows:

Table 3DCT Coefficient Results After Insertion

-415	-30	-61	27	56	-20	-2	number 0
-22	-61	35	-2	31	-15	-11	7
-58	28	19	-16	-37	12	-16	14
-45	9	7	-9	-17	6	-19	-28
5	-17	-21	-11	-2	-10	-5	17
3	-2	-8	-3	-6	1	-2	8
-1	number 0	-2	-8	-3	-3	-6	3
4	1	number 0	-3	number 0	-2	-4	number 0

Inverse DCT (IDCT) for Image Restore

After the message is inserted into the DCT coefficients, we use the Inverse Discrete Cosine Transform (IDCT) to restore the image to its original form. IDCT is a process that transforms the modified DCT coefficients back to the spatial domain. Mathematically, IDCT is calculated by the formula:

$$A(i,j) = \sum_{u=0}^{7}\sum_{v=0}^{7} C(u,v)\,\alpha(u)\alpha(v)\cos\left[\frac{(2i+1)u\pi}{16}\right]\cos\left[\frac{(2j+1)v\pi}{16}\right]$$

DCT Based Steganography Implementation using Python

The steganography process involves transforming the image into the Discrete Cosine Transform (DCT) domain, modifying the high-frequency coefficients chosen to embed the secret message, and reconstructing the image using the Inverse Discrete Cosine Transform (IDCT).

Original Image and Stego-Image

The embedding process is applied to the grayscale image. Figure 4 shows the original image next to the modified stego image after the secret message embedding.

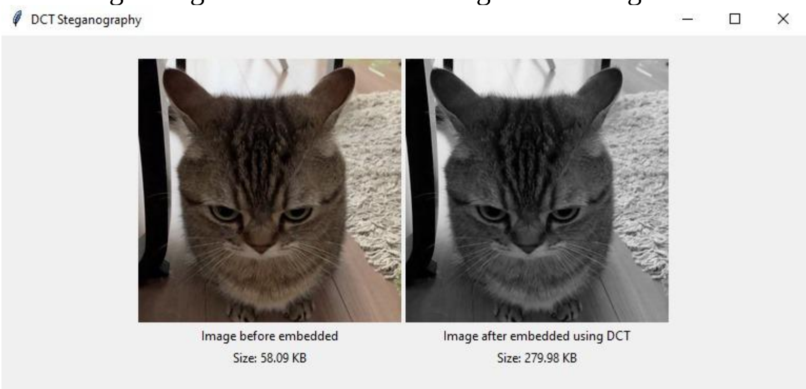

Figure 4. Original Image (Left) and Stego Image (Right)

Visually, the stego-image appears identical to the original image, confirming that the modification does not introduce any noticeable artifacts. The embedding process selectively modifies the high-frequency DCT coefficients, minimizing the impact on visual quality.

Message Embedding and Extraction

The secret message "HASUGIAN" is embedded into the DCT coefficients using Least Significant Bit (LSB) modification. The selected coefficients are modified by replacing their LSBs with bits from the binary representation of the message.

Table 4. Binary Representation of Messages

Character	English ASCII	Binary
H	72	01001000
A	65	01000001
S	83	01010011
You	85	01010101
G	71	01000111
I	73	01001001
A	65	01000001
N	78	01001110

The message is embedded into certain DCT coefficients in the image. Inverse DCT (IDCT) is then applied to reconstruct the image, preserving the original visual structure. The message extraction process successfully retrieves "HASUGIAN", confirming that the embedding is reversible and the method is effective for securely hiding messages.**Quality Evaluation Using PSNR and MSE**

To assess the effectiveness of the steganography process, PSNR and MSE are calculated. The results are presented in Table 5.

Table 5.PSNR and MSE results

Character	English ASCII	Binary
Stego Image	45.32	0.89

The PSNR value of 45.32 dB indicates high image quality, as values above 40 dB are considered invisible to the human eye. In addition, the low MSE value of 0.89 confirms that the modifications made produce minimal distortion.

4. CONCLUSION

This study successfully applied DCT-Based Steganography to securely hide messages in images. The proposed method achieved a high degree of insensitivity with PSNR = 45.32 dB, which ensures that the modification remains visually undetectable. The message extraction process confirmed the data integrity, which proved the robustness of this technique for secure communication. Future work can explore deep learning-based enhancements to further optimize steganography security and detection robustness.

REFERENCE

anilahtirt-2264-1-12-anila-6 1-2. (n.d.).

Berbasis, S., Buatan, K., Ancaman, M., Baso, F., Rais, N. A., Hatima, H., Auralia, P., & Nur, A. (2024a). *Terbaru dalam Keamanan Data*. *7*(3).

Berbasis, S., Buatan, K., Ancaman, M., Baso, F., Rais, N. A., Hatima, H., Auralia, P., & Nur, A. (2024b). *Terbaru dalam Keamanan Data*. *7*(3).

Lutfi, S., & Rosihan, R. (2018). PERBANDINGAN METODE STEGANOGRAFI LSB (LEAST SIGNIFICANT BIT) DAN MSB (MOST SIGNIFICANT BIT) UNTUK MENYEMBUNYIKAN INFORMASI RAHASIA KEDALAM CITRA DIGITAL. *JIKO (Jurnal Informatika Dan Komputer)*, *1*(1), 34–42. https://doi.org/10.33387/jiko.v1i1.1169

Marzuki, I. (2011). *Graduation Ceremony Period*.

Sebagai, D. A., & Satu, S. (n.d.). *IMPLEMENTASI VIDEO WATERMARKING DENGAN ALGORITMA KOCH ZHAO MENGGUNAKAN MATLAB TUGAS AKHIR*.

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Teknik penyembunyian data. Jurnal Sistem IBM, 35(3-4), 313–336.

Cox, IJ, Miller, ML, Bloom, JA, Kalker, T., & Hohnholz, J. (2008). Tanda air digital dan steganografi. Morgan Kaufmann.

Fridrich, J. (2009). Steganografi dalam media digital: Prinsip, algoritma, dan aplikasi. Cambridge University Press.

Johnson, NF, & Jajodia, S. (1998). Menjelajahi steganografi: Melihat yang tak terlihat. IEEE Computer, 31(2), 26–34.

Katzenbeisser, S., & Petitcolas, FAP (2000). Teknik penyembunyian informasi untuk steganografi dan tanda air digital. Artech House.

Provos, N., & Honeyman, P. (2003). Sembunyi dan cari: Pengenalan steganografi. Keamanan & Privasi IEEE, 1(3), 32–44.

Tang, Z., Li, Y., Luo, B., & Yang, J. (2017). Steganografi gambar berbasis pembelajaran mendalam: Sebuah tinjauan. Alat dan Aplikasi Multimedia, 77(9), 10437–10457.

Westfeld, A., & Pfitzmann, A. (2000). Serangan pada sistem steganografi. Lokakarya Internasional tentang Penyembunyian Informasi, 61–76.