Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

A LIGHTWEIGHT ADAPTIVE LSB AND GLOBAL-DCT FRAMEWORK FOR HIGH-FIDELITY IMAGE STEGANOGRAPHY AND COPYRIGHT AUTHENTICATION

Bheem Sen Neel[*1], Noor Ahmed Shaikh[2], Riaz Ahmed Shaikh[3], Aftab Ahmed[4]

[*1,2,3,4]*Institute of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan.*

[*1]bheemmanshani@gmail.com,[2]noor.shaikh@salu.edu.pk,[3]riaz.shaikh@salu.edu.pk,
[4]aftab.baloch69@gmail.com

DOI: https://doi.org/10.5281/zenodo.20342295

Keywords
Image steganography; adaptive LSB; Global-DCT watermarking; copyright authentication; image security; PSNR; SSIM; BER; NCC.

Article History
Received: 11 March 2026
Accepted: 21 April 2026
Published: 22 May 2026

Corresponding Author: *
Bheem Sen Neel

Abstract
Embedding hidden information within digital images is challenging, as the image should be visually clean while the hidden information should be recoverable. In this study, a light weight image hiding framework is proposed which is based on the combination of adaptive Least Significant Bit (LSB) steganography and Global Discrete Cosine Transform (Global-DCT) watermarking. The method is based on concealing an encrypted payload in edge and texture sensitive pixel regions and embedding a small copyright/authentication watermark in some of the pairs of Global-DCT coefficients. The proposed method was tested on 5,524 images from Kodak, Uncompressed Color Image Database (UCID) and BOSSBase/BOWS2 datasets. The results indicate that the visual quality is high with an average Peak Signal-to-Noise Ratio (PSNR) of 47.386 dB and Structural Similarity Index Measure (SSIM) of 0.995079. The encrypted payload was extracted with 100% cleanliness, and the compact watermark got a clean Bit Error Rate (BER) of 0.01668 and Normalized Correlation (NCC) of 0.98596. The watermark was found to be robust against blur, resizing, JPEG Q95/Q90 compression and Gaussian noise under image processing attacks. The results demonstrate that the proposed framework is a practical trade-off between high fidelity steganography and low overhead copyright authentication. Simple, reproducible, and not requiring deep model training, it can be used for secure image sharing and ownership verification applications.

1. INTRODUCTION

Digital images are utilized for social media, e-commerce, education, healthcare, cloud storage and official communication. This extensive use of images has resulted in two key security requirements. First, there are times when private information must be transmitted without the knowledge that a secret message is being transmitted. Second, once an image is shared online, the owner of the image may be required to establish copyright, authorship, or authenticity. Cryptography can be used to secure the content of a message, but it cannot obscure the existence of the message. To make up for this, steganography and watermarking are used to embed information within digital media. In image steganography, secret information is embedded within a cover image and the resulting image, called stego image, should be almost identical to the cover image. The criteria for a good steganography method are: high visual quality, reliable message recovery, and adequate embedding capacity. One of the simplest and most common image

steganography techniques is called Least Significant Bit (LSB) substitution. It conceals data by altering the least significant bits in the pixels. These bit changes are very small, and LSB methods can result in high PSNR and SSIM values. Simple LSB methods are however generally vulnerable to compression, resizing, filtering and attacks [1] [2] [7].

In recent studies, LSB based steganography has been enhanced by making the embedding process adaptive. Adaptive methods choose textured areas, areas with edges, or areas that are not as visually sensitive to embed data. This minimizes the visible distortion and enhances security by hiding changes in less conspicuous areas. Recently, content-adaptive approaches have been proposed, such as saliency fusion [2], ant colony optimization [4] and hybrid encryption [5] to enhance imperceptibility and extraction reliability. Other studies use LSB in conjunction with deep learning or compression techniques to enhance security and payload management [1], [3], [6]. Digital watermarking has a related but different purpose. In watermarking, the hidden information is typically smaller, and is applied to protect copyright, to verify authenticity, to identify ownership, or to detect tampering. A watermark should be perceptible after common image processing operations. To this end, powerful watermarking techniques are frequently based on transform domain methods like the Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Redundant Discrete Wavelet Transform (RDWT), Integer Wavelet Transform (IWT), and Singular Value Decomposition (SVD). Such methods involve the watermarking of frequency or transformed coefficients, instead of the actual image pixels, which can enhance the watermark's resistance to image processing attacks [8-11]. These techniques are based on embedding the watermark in the frequency coefficients instead of the pixel values, which can enhance the watermark's resistance to image processing attacks [8-11].

The balance between imperceptibility, robustness and capacity is still the major problem in recent watermarking research. Very strong watermark can withstand attacks, but it can also lower image quality. A very light watermark might maintain the image quality but might not be effective after compression or noise. Another popular watermarking method is deep learning based watermarking, where neural networks can learn strong embedding patterns. But deep learning approaches are usually expensive in terms of training time, computational power and the need for large training sets [8-11]. However, classical transform domain approaches are still relevant as they are simple, understandable and easy to replicate. There are many approaches that are already available, but they are only based on either steganography or watermarking and not both in one simple framework. Pure LSB steganography is able to embed more information with high image quality, but it is not robust when compressed. Pure watermarking can be more robust, but typically contains a limited amount of information.

Hence, a light weight framework is required that can be used for hiding secure payloads and also for copyright/authentication watermarking, while introducing low visual distortion. In this regard, this study introduces an Adaptive LSB and Lightweight Global-DCT Framework for High-Fidelity Image Steganography and Robust Copyright Authentication. The proposed method is based on two cascaded embedding operations. Firstly, the secret payload is compressed and encrypted and inserted into the image by adaptive LSB. The adaptive LSB layer identifies the textured and edge-sensitive areas, making the changes less noticeable. Secondly, a small copyright/authentication watermark is embedded by Global-DCT coefficient-pair modification.

The Global-DCT layer adopts the selected pairs of frequency domain coefficients to enhance the watermark detectability after common image processing operations. The proposed method was tested on 5,524 images obtained from three image sources: Kodak, UCID and BOSSBase/BOWS2. The Kodak set is a common standard visual-quality test set [22]. UCID is a database of 1,338 uncompressed colour images, which is commonly used in image processing and image retrieval research [23]. BOSSBase 1.01 is a standard benchmark in the research of steganography and steganalysis, and it consists of 10,000 grayscale images from the BOSS competition [21, 24]. BOWS2 also includes 10,000 gray level images and is widely

used in watermarking and steganography studies [25]. The final dataset of this experiment comprised 24 Kodak images, 500 UCID images and 5,000 BOSS/BOWS2 images, totaling 5,524 images.

The final results demonstrate that the proposed method has been able to obtain high visual quality and reliable clean-channel extraction. Across 5,524 images, the average PSNR was 47.386 dB, the average SSIM was 0.995079, and the average MSE was 1.4461. The LSB payload extraction success rate was 100% and the average adaptive LSB capacity was 0.393 bpp. The average BER of the clean watermark extraction was 0.01668 and the average NCC was 0.98596. The results demonstrate that the proposed method can maintain high image quality in the case of embedding an encrypted payload and a small authentication watermark.

The main contributions of this study are:

1. This paper proposes a lightweight image hiding framework by combining the adaptive LSB steganography with compact Global-DCT watermarking.
2. The method includes an encrypted secret payload and a copyright/authentication watermark in the same stego image.
3. Embedding data in regions of an image that have texture and edge information using the adaptive LSB layer can enhance imperceptibilit.
4. The Global-DCT layer is the one that embeds a small watermark of 16 bits by modifying the coefficient pairs.
5. Evaluation of the method on 5,524 images from Kodak, UCID and BOSSBase/BOWS2.
6. The method has an average PSNR of 47.386 dB, an average SSIM of 0.995079 and 100% clean payload extraction.
7. The method is tested for JPEG compression, Gaussian noise, salt and pepper noise, resizing, blur, sharpening and cropping.

The rest of the paper is organized as follows. The related work on adaptive LSB steganography, deep-learning steganography and transform-domain watermarking is reviewed in Section 2. The proposed adaptive LSB and lightweight Global-DCT method is explained in Section 3. The datasets, attack settings, and evaluation metrics are presented in Section 4. Section 5 discusses the experimental results using tables and figures. The paper ends with suggestions for future improvements in Section 6.

2. RELATED WORK

Digital image hiding has been evolved in two major directions: image steganography and image watermarking. The primary difference between steganography and watermarking is that watermarking is concerned with embedding ownership, copyright, and authentication data, whereas steganography is primarily concerned with hiding the secret information within an image. The common problem in both areas is that the information must be hidden without causing perceptible distortion. Meanwhile, the embedded data should be recoverable during normal use or image processing operations.

2.1 Image Steganography

The image steganography techniques are generally classified into three categories: spatial domain, transform domain and learning-based. Spatial domain methods directly manipulate the pixels in the image. The most popular one is Least Significant Bit (LSB) substitution, in which the secret bits are embedded into the least significant bits of the pixels. LSB methods are simple and often generate high quality images since they only change the value of the image at the pixel level. Simple LSB embedding is, however, fragile as compression, resizing, filtering or noise can destroy the hidden bits [7]. Recently, the embedding process has been made adaptive, which has enhanced LSB steganography. Adaptive methods choose regions where data can be hidden without being noticed in the same manner. These areas are typically edges, textured areas, or visually complex areas of the image. Aljughaiman et al. [2] proposed a content-adaptive LSB method with saliency fusion, ant colony optimization and hybrid encryption. They have demonstrated that the embedding position can be adaptively selected for better imperceptibility and security. In the same way, Alrawashdeh et al. [4] applied edge guided attention maps and a convolutional neural network to guide adaptive embedding. These studies confirm the notion that embedding LSB can be more effective with the

help of edge and texture information. With the help of deep learning, image steganography has also been enhanced. Song et al. [1] surveyed deep learning based image steganography and demonstrated that neural models can learn complex embedding patterns. Sanjalawe et al. [3] suggested a multi-layered steganographic approach using deep learning for better data security. Other recent studies also fuse deep learning with adaptive embedding to enhance the hiding capacity, visual quality, and detection resistance [5,6]. These approaches are effective, but they can be expensive in terms of training, data and computation. This means that lightweight methods remain valuable, particularly if the desired system is to be reproduced and has a low complexity..

2.2 Image Watermarking

Watermarking of images is similar to steganography, but for a different purpose. Watermarking is typically used for ownership verification, copyright protection, authentication, and tamper detection, and the embedded data is typically smaller. A good watermarking technique should maintain the quality of the image and be able to extract the watermark after the image is attacked by image processing techniques. Spatial-domain watermarking is easy to implement, but is typically vulnerable to attacks. For this reason, many powerful watermarking techniques are based on transform domain techniques. The commonly used transform domain techniques are Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Redundant Discrete Wavelet Transform (RDWT), Integer Wavelet Transform (IWT), and Singular Value Decomposition (SVD). These methods do not directly modify the pixels, but rather the transformed coefficients in which the watermark is embedded. This can enhance the robustness because the transformed coefficients might be more stable during compression, resizing and filtering [8], [9], [10], [11].

From the recent watermarking survey, it is observed that watermarking in images is still an active research field. Hosny et al. [8] discussed the watermarking using deep learning and provided the explanation that the watermarking of the present era should be imperceptible, robust, and have a high capacity. Challenges and opportunities of deep learning based watermarking were discussed by Ben Jabra et al. [9]. Luo et al. [10] summarized the strong deep image watermarking methods, and Bistroń et al. [11] presented a more recent survey of deep learning watermarking for images. These surveys reveal that learning-based watermarking is on the rise, while classical transform-domain watermarking is still significant due to its explainability and lack of training. There are several recent approaches that use several transforms to increase robustness. Dong et al. [12] presented an adaptive robust watermarking scheme based on chaotic mapping and hybrid transform domain embedding. Chaudhary et al. [13] proposed a hybrid DWT-HMD-SVD watermarking method for medical images. Alrammahi et al. [14] used DWT and RDWT with Mobius transformations and Shubuh et al. [16] used hybrid IWT-DCT-SVD method. These works demonstrate that hybrid transform-domain embedding can be beneficial for robustness, but can also lead to higher algorithmic complexity.

2.3 Steganalysis and Benchmark Datasets

The importance of the steganalysis studies is that they test the detection of hidden information. Many datasets like BOSSBase and BOWS2 are used for steganography and steganalysis studies. Bas et al. [21] proposed the BOSS competition framework that contributed to the standardization of steganographic evaluation. BOSSBase 1.01 is a 10,000 image grayscale database and is commonly used in steganalysis experiments [24]. BOWS2 is also widely used as a grayscale image benchmark in the research of steganography and watermarking [25]. These datasets are still being used for benchmarking in recent steganalysis studies. Denemark et al. [20] suggested selection-channel-aware rich models for steganalysis, which demonstrated that the location of embedding changes is important. Huang et al. [26] investigated content selection for steganalysis and Zhang et al. [27] investigated image source selection in multi-image steganography. These studies confirm the need to test the image hiding methods on a variety of image sets rather than a limited set of images.

In addition to steganography benchmarks, natural color image datasets are also important for visual-quality testing. The Kodak image suite

is a popular small image-processing benchmark [22]. The Uncompressed Color Image Database (UCID) is a database of 1,338 color images and is suitable for image quality assessment for various natural images [23]. Hence, the proposed method is tested on color and grayscale style benchmark images using the Kodak, UCID and BOSSBase/BOWS2 image sources.

2.4 Research Gap

The literature reviewed indicates that the current approaches typically address one aspect of the problem. Adaptive LSB steganography methods are highly imperceptible and have a large capacity, however, they are fragile when compressed or processed on images [2], [4] and [7]. Strong watermarking techniques make the watermark more resistant to attacks, but they can only carry a small amount of information, and stronger watermarks can also degrade image quality [8], [11], [13], [14]. Deep learning approaches can enhance performance, but they are not lightweight and need to be trained [1], [3], [8], [9]. This leaves an opportunity for a straightforward and replicable model that integrates both functions. This framework should not only be able to conceal an encrypted payload with good visual quality but also include a small copyright/authentication watermark which is robust to common image processing attacks. To overcome this, the proposed method is based on the adaptive LSB embedding technique and lightweight Global-DCT coefficient-pair watermarking technique.

Table 1. Summary of Related Work and Research Gap

Study	Main technique	Strength	Limitation / gap
Song et al. [1]	Deep learning-based steganography survey	Covers recent neural steganography methods	Deep models often require training and higher computation
Aljughaiman et al. [2]	Adaptive LSB with saliency, ACO, and encryption	Improves adaptive embedding and security	Mainly focused on steganography rather than robust watermarking
Sanjalawe et al. [3]	Deep learning-driven multi-layer steganography	Improves hidden-data security	More complex than lightweight classical methods
Alrawashdeh et al. [4]	Edge-guided adaptive steganography	Uses edge attention to improve embedding	Requires additional feature/attention processing
Alanzy [7]	LSB with hybrid encryption	Simple and secure payload embedding	Spatial embedding remains fragile under attacks
Hosny et al. [8]	Deep image watermarking survey	Explains modern watermarking challenges	Survey does not provide a lightweight hybrid implementation
Ben Jabra et al. [9]	Deep learning watermarking review	Discusses robustness and learning-based watermarking	Deep methods may be computationally expensive
Dong et al. [12]	Chaotic mapping and hybrid transform watermarking	Improves robustness using transform domain	More complex than compact DCT-only watermarking
Chaudhary et al. [13]	DWT-HMD-SVD watermarking	Strong medical image watermarking approach	Hybrid transforms can increase complexity
Alrammahi et al. [14]	DWT and RDWT watermarking	Improves robustness under attacks	Focuses on watermarking, not encrypted payload hiding
Shubuh et al. [16]	IWT-DCT-SVD watermarking	Uses multiple transforms for robustness	Higher algorithmic complexity

Proposed method	Adaptive LSB + lightweight Global-DCT watermarking	Hides encrypted payload and compact authentication watermark in one framework	Cropping remains a limitation without synchronization

3. PROPOSED METHOD

The proposed method is a combination of adaptive spatial domain steganography and lightweight transform domain watermarking. The primary goal is to hide an encrypted secret payload and at the same time embed a compact copyright/authentication watermark in the same image. The method is intended to minimize the distortion in the image and to enable the verification of the watermark after common image processing attacks. The final method is named Proposed Adaptive LSB + Lightweight Global-DCT Watermarking. The final implementation is with a 16-bit watermark, 3 repetitions, alpha = 0.75, mid_high_quality frequency profile, and pair gap = 1. The final settings were applied to the entire experiment of 5,524 images.

3.1 Overview of the Proposed Framework

There are two interrelated embedding stages in the proposed framework. Firstly, the copyright/authentication watermark is hidden in the transformed image by the Discrete Cosine Transform (DCT). Second, the encrypted secret payload is planted in a predetermined position of the LSB (Least Significant Bit) in an adaptive way. The watermark is applied first, which alters global frequencies, and the LSB layer is applied next, which involves very small changes at the pixel level to add the encrypted payload. The watermark is embedded first because it modifies global frequency coefficients, while the LSB layer is added afterward to insert the encrypted payload with very small pixel-level changes.

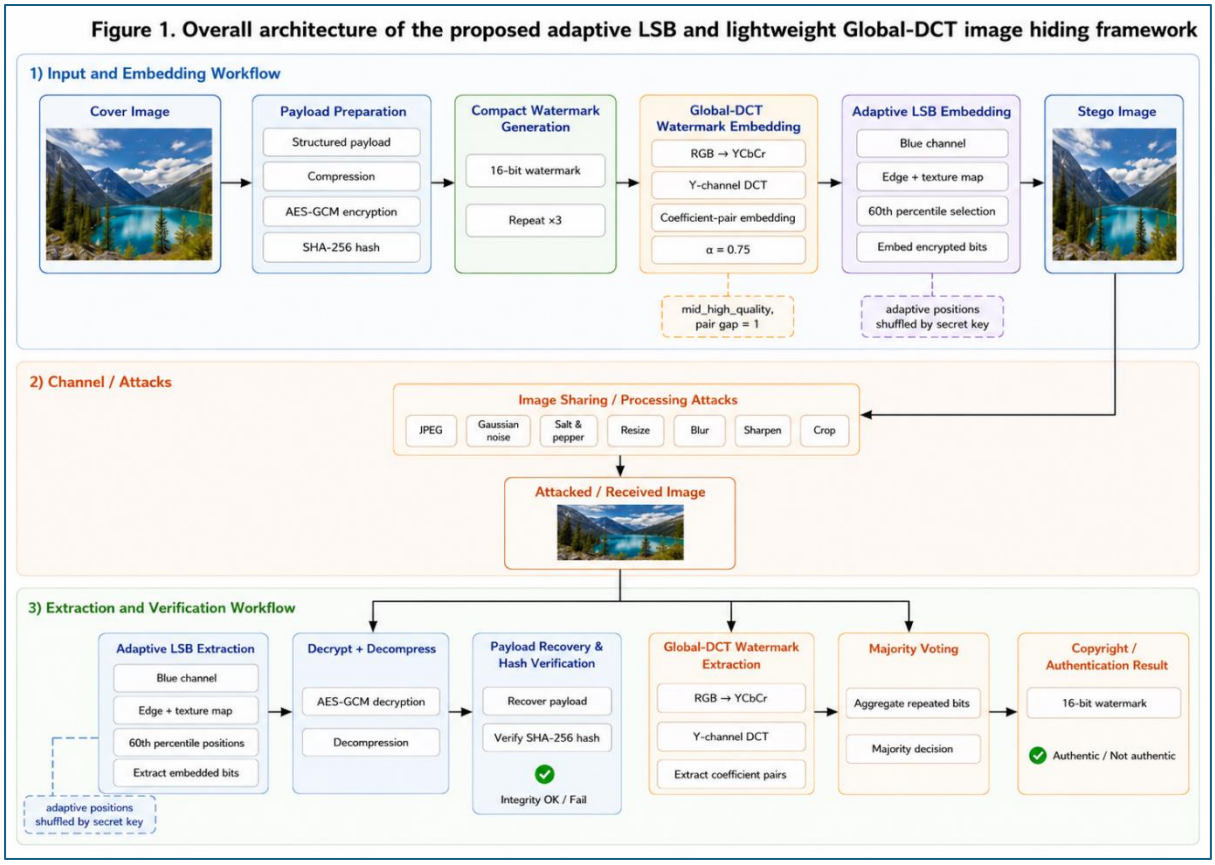


Figure 1. Overall architecture of the proposed adaptive LSB and lightweight Global-DCT image hiding framework

The figure illustrates the cover image, encrypted payload, compact copyright/authentication watermark, embedding path of the Global-DCT, adaptive embedding path of LSB, final stego image, attack channel, and extraction process. The framework can be summarised as follows:

1. Cover image is transformed into an appropriate image representation.
2. Compact watermark is created from ownership/authentication information.
3. Selected pairs of Global-DCT coefficients are watermarked.
4. Secret payload is compressed and encrypted.
5. Adaptive LSB position decisions based on the edge and texture information.
6. Payload bits of the encrypted information are embedded at the chosen LSB positions.
7. Final Stego image is generated.
8. Payload is recovered from adaptive LSB locations and the watermark is recovered from pairs of Global-DCT coefficients during extraction.

This design maintains simplicity and repeatability of the method. It does not need extensive model training, but it still has two beneficial features: high fidelity payload hiding and light watermark authentication.

3.2 Payload Preparation

The secret payload is ready prior to embedding. The payload includes owner information, hidden message, timestamp, method name and purpose. An authentication hash is also created using a Secure Hash Algorithm 256-bit (SHA-256) to check if the payload that was extracted is correct. The uploaded final payload file reveals that the payload contains the method name, owner identity, purpose, and authentication hash (SHA-256).

The payload preparation process has three steps:

1. Payload information is transformed to structured text format.
2. Payload is compressed to decrease the number of bits.
3. Compressed payload encrypted with Advanced Encryption Standard in Galois/Counter Mode (AES-GCM).

Encryption is significant because even if someone finds out that hidden data is present, it will be protected. Later, the hash is used to verify if the payload was extracted correctly.

Let the original payload be represented as:

$$P$$

The compressed payload is:

$$P_c = Compress(P)$$

The encrypted payload is:

$$P_e = Encrypt(P_c, K)$$

where $\boldsymbol{K}$ is the secret key. The encrypted payload $\boldsymbol{P_e}$ is then converted into a binary bitstream before LSB embedding.

3.3 Lightweight Global-DCT Watermark Generation

The watermark is a compact binary sequence used for the purpose of copyright or authentication verification. The final methods is to set the watermark length to 16 bits. The watermark is created based on the owner information, secret key and method-specific identifier. A compact watermark is used because fewer coefficients have to be changed, which will help to minimize the distortion of the image.

The watermark bitstream is represented as:

$$W = \{w_1, w_2, w_3, \dots, w_n\}$$

where:

$$n = 16$$

Each watermark bit is repeated three times to improve extraction reliability:

$$W_r = Repeat(W, r)$$

where:

$$r = 3$$

Therefore, the total number of embedded watermark decisions is:

$$16 \times 3 = 48$$

This is a light-weight effort. Previous pilot tests indicated that the more watermark is repeated, the more robust it is, but the less visually pleasing it is. The 16-bit and 3-repetition setting has been selected, which offers a good

compromise between image quality and watermark recovery.

3.4 Global-DCT Coefficient-Pair Watermark Embedding

The watermark is hidden in the brightness part of the image. The image is first converted from Red-Green-Blue (RGB) colour space to Y-Cb-Cr (YCbCr) colour space. Y is the luminance channel and Cb and Cr are the chrominance channels. The watermark is hidden in the Y channel since luminance carries significant structural information and is more resistant to common image processing operations.

The two-dimensional Discrete Cosine Transform is applied to the Y channel:

$$D = DCT(Y)$$

where $\boldsymbol{D}$ is the DCT coefficient matrix.

The method chooses the coefficient pairs in a mid-to-high quality frequency region. This area is away from very low frequency coefficients that can lead to visible distortion, and away from extremely high frequency coefficients that can be easily thrown out by compression. The profile that is selected in the final implementation is named mid_high_quality.

For each repeated watermark bit, one DCT coefficient pair is selected:

$$(D_a, D_b)$$

The embedding rule is based on ordering of the coefficients. When the watermark bit is 1, the first coefficient should be at least $\boldsymbol{\alpha}$. (embedding strength) larger than the second coefficient. When the watermark bit is 0, the second coefficient should be at least α larger than the first coefficient.

The final embedding strength is:

$$\alpha = 0.75$$

For watermark bit 1:

$$D_a - D_b \geq \alpha$$

For watermark bit 0:

$$D_b - D_a \geq \alpha$$

If the required condition is not satisfied, the two coefficients are slightly adjusted. The adjustment is split between the two coefficients to minimize distortion that can be seen.

After watermark embedding, the inverse DCT is applied:

$$Y' = IDCT(D')$$

The modified luminance channel $\boldsymbol{Y'}$ is mixed with the original Cb and Cr channels and the image converted back to RGB.

3.5 Adaptive LSB Position Selection

Watermark embedding is followed by the embedding of the encrypted payload using adaptive LSB embedding. The method does not embed payload bits in random or sequential pixels, but rather in textured and edge sensitive regions. This is done because the small pixel changes are not as noticeable in the visually active regions as in the smooth regions.

The adaptive selection uses two indicators:

1. Edge strength
2. Local texture variation

The channel that is used for LSB embedding is the blue channel. The method is designed to make the position selection stable, not depending directly on the LSB values, but using the upper bits of the selected channel. This will keep the embedding operation from altering the selection map significantly. The edge score is calculated by edge detector and the local variation is calculated by a small neighborhood window. The two values are combined in an embedding suitability score:

$$S(x, y) = E(x, y) + V(x, y)$$

where $\boldsymbol{E(x, y)}$ is the edge score and $\boldsymbol{V(x, y)}$ is the local texture variation at pixel position $\boldsymbol{(x, y)}$.

A threshold is selected using the 60th percentile:

$$T = Percentile(S, 60)$$

Pixels with scores greater than or equal to the threshold are selected as candidate embedding positions:

$$M(x,y) = \begin{cases} 1, & S(x,y) \geq T \\ 0, & otherwise \end{cases}$$

To minimize boundary related artifacts, border pixels are not included. A secret-key based random seed is then used to shuffle the selected positions. This makes it hard to guess where the payload is, unless you have the right key.

3.6 Adaptive LSB Payload Embedding

The encrypted payload is converted to binary bitstream. A 32-bit header containing the length of the payload is added before embedding. This enables the extraction process to determine the number of bits that need to be recovered.

Let the encrypted payload bitstream be:

$$B = \{b_1, b_2, b_3, .., b_m\}$$

The 32-bit length header is:

$$H$$

The complete embedded bitstream is:

$$B_f = H \,||\, B$$

where || represents concatenation.

For each selected pixel position, the least significant bit is replaced by one payload bit:

$$p' = p \,\&11111110_2 \,) \,|\, b_i$$

where $\boldsymbol{p}$ is the original pixel value, $\boldsymbol{p'}$ is the modified pixel value, and $\boldsymbol{bi}$ is the payload bit to be embedded. This operation changes each selected pixel value by at most one intensity level. The positions are taken from textured and edge sensitive areas, so the visual effect is very small.

3.7 Extraction Process

The extraction process undoes the embedding steps. First, the adaptive LSB positions are re-generated with the same secret key, channel and percentile of the texture. The first 32 bits extracted are used to recover the payload length. Then the necessary number of payload bits are taken out and converted back to encrypted bytes.

The encrypted payload is then decrypted and decompressed:

$$P_c = Decrypt(P_e, K)$$
$$P = Decompress(P_c)$$

The hash of the payload is extracted and compared with the original authentication hash (SHA-256). If the hashes are the same, the payload extraction is deemed successful.

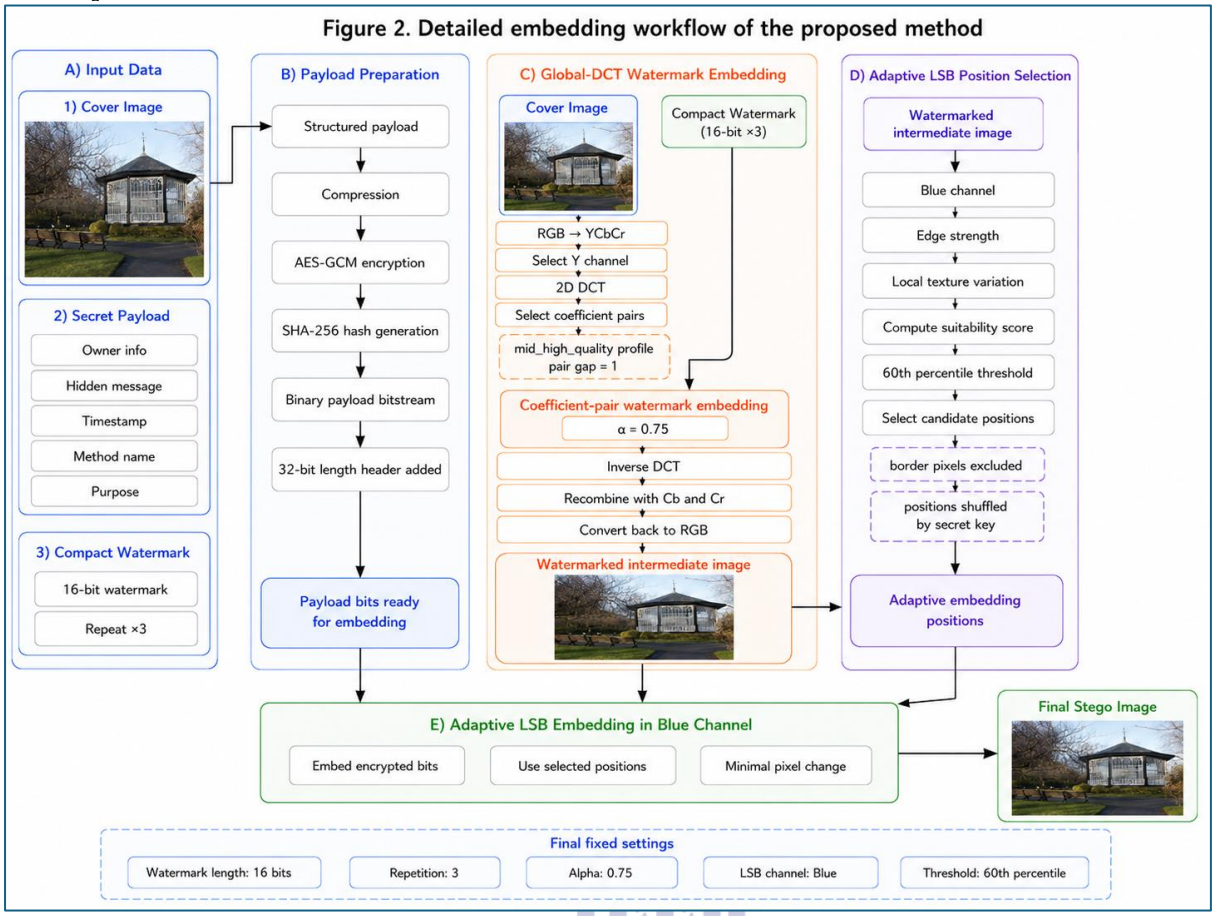
To extract watermark, the stego or attacked image is first transformed to YCbCr format and then DCT is performed on the luminance channel. The same coefficient pairs are chosen in the same way, based on the same secret-key.

For each pair, the extracted bit is decided by comparing the two coefficients:

$$\hat{\omega}_i = \begin{cases} 1, & D_a > D_b \\ 0, & D_a \leq D_b \end{cases}$$

Each watermark bit is repeated three times and the final watermark bit is recovered using majority voting. This makes it more reliable when the coefficient is slightly changed due to compression, resizing or noise.

3.8 Proposed Method Workflow



The workflow shows payload compression and encryption, generation of the SHA-256 hash, embedding of the lightweight Global-DCT coefficient-pair watermark, adaptive LSB position selection using the edge and texture information, and final stego image generation. The entire embedding process is presented in Algorithm 1.

Algorithm 1. Proposed embedding algorithm

Input: Cover image I, secret payload P, owner/authentication information O, secret key K
Output: Stego image I_s

1. Convert payload P into structured text.
2. Generate SHA-256 authentication hash for the payload.
3. Compress the payload.
4. Encrypt the compressed payload using AES-GCM and secret key K.
5. Generate a 16-bit watermark W from ownership/authentication information.
6. Repeat each watermark bit three times.
7. Convert cover image I from RGB to YCbCr.
8. Apply DCT to the Y channel.
9. Select Global-DCT coefficient pairs from the mid_high_quality frequency region.
10. Embed repeated watermark bits using coefficient-pair ordering and $\alpha = 0.75$.
11. Apply inverse DCT and reconstruct the watermarked image.
12. Select adaptive LSB embedding positions using edge and texture scores.
13. Add a 32-bit payload-length header to the encrypted payload bitstream.
14. Embed payload bits into selected LSB positions.
15. Save the final stego image I_s.

Algorithm 2. Proposed extraction algorithm

Input: Stego or attacked image I_s, secret key K
Output: Extracted payload $\hat{P}$, extracted watermark $\hat{W}$

1. Regenerate adaptive LSB positions using secret key K.
2. Extract the first 32 bits to obtain payload length.
3. Extract the encrypted payload bitstream from selected LSB positions.
4. Decrypt the payload using AES-GCM and secret key K.
5. Decompress the decrypted payload.
6. Verify the extracted payload using SHA-256 hash comparison.
7. Convert image to YCbCr.
8. Apply DCT to the Y channel.
9. Select the same Global-DCT coefficient pairs.
10. Compare each coefficient pair to extract repeated watermark bits.
11. Use majority voting to recover the final 16-bit watermark.
12. Compute Bit Error Rate (BER) and Normalized Correlation (NCC) between original and extracted watermark.

3.9 Main Design Advantages

The proposed method has four major merits. First, it is a combination of payload hiding and authentication watermarking in a single framework. The adaptive LSB layer conceals the encrypted secret payload and the Global-DCT layer incorporates a small watermark for ownership or authentication. Secondly, it is lightweight. No training of a deep neural network is required. It employs deterministic operations like compression, encryption, adaptive pixel selection, DCT and coefficient-pair modification. Thirdly, the watermark is small. A total of 16 watermark bits are used, with

each bit repeated three times. This reduces the number of DCT coefficient pairs to be modified, and helps maintain image quality. Fourth, the method is reproducible. All the important parameters are set: watermark bits = 16, repetitions = 3, alpha = 0.75, frequency profile = mid_high_quality, pair gap = 1, and 60th percentile threshold is used for adaptive LSB selection. The same conditions were applied in the final full dataset experiment.

4. EXPERIMENTAL SETUP

This section presents the details of the dataset, implementation environment, final parameter settings, attack conditions and evaluation metrics for testing the proposed method. The aim of the experiment was to evaluate the image quality and hidden-information recovery. Visual distortion metrics were used to measure image quality and payload extraction success and watermark similarity metrics were used to measure recovery performance.

4.1 Dataset Description

The proposed method was tested on a mixed data set of 5,524 images from Kodak, Uncompressed Color Image Database (UCID) and a sampled BOSSBase/BOWS2 image set. The use of multiple datasets was to try the method with a variety of image types rather than a single small benchmark. A small standard set for visual-quality testing is provided by Kodak [22]. UCID is a natural color image and is useful for image quality assessment for different scenes [23]. BOSSBase and BOWS2 are popular in steganography and steganalysis research as they are benchmark grayscale image sources [21], [24], [25].

Table 2. Dataset distribution used in the final experiment.

Dataset	Number of images	Image type	Purpose in experiment
Kodak	24	Color images	Standard visual-quality benchmark
UCID	500	Color images	Natural scene evaluation
BOSSBase/BOWS2 sample	5,000	Grayscale-style benchmark images	Large-scale steganography/watermarking evaluation
Total	**5,524**	Mixed	Final full-dataset evaluation

Table 2 indicates that the 24 Kodak images were not the only ones used in the final experiment. Kodak was primarily used as a benchmark for visual quality first, and then UCID and BOSSBase/BOWS2 are used to expanded the testing variety and volume. The large BOSSBase/BOWS2 sample was used to assess the method on a steganography-style benchmark, and the UCID images were used to assess performance on color natural scenes. This distribution of datasets strengthens the evaluation, as opposed to a small single-dataset experiment.

4.2 Final Method Parameters

The last proposed method employed a single configuration for all images. No modes were reported in the final paper. It is a combination of adaptive Least Significant Bit (LSB) payload embedding and lightweight Global Discrete Cosine Transform (Global-DCT) watermarking. The final implementation employed a compact 16-bit watermark, which was repeated for three times for majority-based recovery of the images. The embedding strength of the watermark was chosen as $\alpha = 0.75$. The frequency profile chosen was mid_high_quality and the gap between coefficient pairs was 1. The settings were selected because previous pilot experiments indicated that the heavier the watermark the more robust the watermark, but the lower the visual quality. The final setting is thus centered on the lightness of the imperceptibility and the robustness..

Table 3. Final parameter settings of the proposed method.

Parameter	Final value	Purpose
Watermark length	16 bits	Compact copyright/authentication watermark
Watermark repetition	3	Improves recovery through majority voting
Embedding strength alpha	0.75	Controls DCT coefficient-pair modification
Frequency profile	mid_high_quality	Balances visibility and robustness
Pair gap	1	Selects nearby DCT coefficient pairs
LSB channel	Blue channel	Used for encrypted payload embedding
Adaptive LSB threshold	60th percentile	Selects edge/texture-sensitive regions
Payload protection	AES-GCM encryption + SHA-256 hash	Protects and verifies hidden payload

Table 3 shows that the final method was tested with a reproducible and fixed setting as shown in table 3. The watermark is 16-bit, which results in small transform-domain changes. A repetition value of 3 provides limited redundancy with not too much distortion. The alpha value of 0.75 prevents strong DCT coefficient modification, thus ensuring high Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

4.3 Payload and Watermark Preparation

The secret payload contained owner information, method description, purpose, timestamp, and an authentication hash. The payload was first compressed and then encrypted using Advanced Encryption Standard in Galois/Counter Mode (AES-GCM). After extraction, the correctness of the payloads was verified by computing a Secure Hash Algorithm 256-bit (SHA-256) hash. The uploaded payload file verifies that the payload contains owner information, method name, purpose, and a SHA-256 authentication hash. The watermark was created based on the ownership and authentication information. It was transformed to a 16-bit binary watermark. The watermark bits were each repeated three times, resulting in 48 watermark embedding decisions. This small size was chosen to minimize distortion while enabling watermark verification following multiple attacks.

The final stego image thus had two types of hidden information:

1. An encrypted payload embedded through adaptive LSB.
2. A compact copyright/authentication watermark embedded through Global-DCT coefficient-pair modification.

4.4 Attack Setup

The proposed method was tested in clean and attacked conditions. The clean condition was when the extraction was done from the stego image without any external attack. The attack conditions tested the survivability of the watermark under typical image processing techniques.

The following attacks were used:

Table 4. Image processing attacks used for robustness testing.

Attack type	Setting	Purpose
No attack	PNG stego image	Clean extraction baseline
JPEG compression	Q95, Q90, Q80, Q70, Q60	Tests compression robustness
Gaussian noise	$\sigma = 2$, $\sigma = 3$	Tests noise resistance
Salt-and-pepper noise	0.002 amount	Tests impulse noise resistance
Resize attack	0.90 and 0.75 scale round-trip	Tests scaling robustness
Gaussian blur	Radius = 1	Tests filtering robustness
Sharpening	Standard sharpen filter	Tests enhancement robustness

Crop and restore	0.95 and 0.90 center crop	Tests geometric distortion sensitivity

Table 4 indicates that the experiment includes both mild and more powerful image processing operations. JPEG Q95 and Q90 are common high quality compression settings, JPEG Q80, Q70 and Q60 are stronger compression settings. Common operations in image sharing and editing are resizing, blurring, adding noise, and sharpening. Crop-and-restore attacks were also tried, however it was not expected to perform well as there are no synchronization markers or geometric alignment in the proposed method.

4.5 Evaluation Metrics

Both image-quality metrics and recovery metrics were used in the experiment. The image-quality measures were used to determine the closeness of the stego image to the cover image. Recovery metrics were used to determine the correct extraction of the hidden payload and watermark.

4.5.1 Mean Squared Error

The Mean Squared Error (MSE) is the average of the squares of the difference between the cover image and the stego image. The lower the MSE the less distortion.

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

where $\boldsymbol{I}$ is the cover image, $\boldsymbol{I_s}$ is the stego image, and $\boldsymbol{M \times N}$ is the image size.

4.5.2 Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) is a measure of distortion between two images in decibels. The higher the PSNR, the better the image quality.

$$PNSR = 10 \log 10 \left(\frac{255^2}{MSE}\right)$$

In image hiding research, a higher PSNR value typically indicates that the stego image is closer to the cover image.

4.5.3 Structural Similarity Index Measure

The Structural Similarity Index Measure (SSIM) is a metric for quantifying the similarity between two images. Structural Similarity Index Measure (SSIM) is a metric that quantifies the similarity of images in terms of luminance, contrast, and structural information. PSNR is based primarily on pixel level error, whereas SSIM is closer to human visual perception. An SSIM value close to 1 means that the stego image is structurally very similar to the original image.

4.5.4 Bit Error Rate

Bit Error Rate (BER) is the ratio of the number of wrongly extracted watermark bits to the total number of bits.

$$BER = \frac{Number\ of\ incorrect\ bits}{Total\ number\ of\ bits}$$

The lower the BER, the better the watermark is recovered. The BER of 0 indicates that there is no extraction loss.

4.5.5 Normalized Correlation

Normalized Correlation (NCC) is used to measure the similarity between the original watermark and the extracted watermark. The greater the NCC, the more similar.

$$NCC = \frac{\sum W(i)\, \widehat{W}(i)}{\sqrt{\sum W(i)^2 \sum \widehat{W}(i)^2}}$$

where $\boldsymbol{W}$ is the original watermark and $\widehat{\boldsymbol{W}}$ is the extracted watermark.

4.5.6 LSB Payload Extraction Success

Payload extraction success indicates if the encrypted payload was extracted and verified with the SHA-256 authentication hash. If the decryption of the extracted payload is successful and the hash is correct, then the extraction is considered successful.

4.6 Experimental Outputs

The implementation resulted in numerical tables and visual figures. The embedding results per image, attack results, and summary results are contained in the main CSV files. The final summary is as follows: 5524 images, average PSNR = 47.386 dB, average SSIM = 0.995079, average MSE = 1.4461, 100% clean LSB extraction, average LSB capacity = 0.393 bpp, average clean watermark BER = 0.01668, and average clean watermark NCC = 0.98596.

5. RESULTS AND DISCUSSION

The experimental results of the proposed adaptive Least Significant Bit (LSB) and lightweight Global Discrete Cosine Transform (Global-DCT) image hiding framework are presented in this section. The discussion is split in four sections: imperceptibility of the clean channel, payload and watermark recovery, qualitative visual analysis, and robustness against image processing attacks. The last experiment was conducted on 5,524 images with fixed parameter setting as mentioned in Section 4. The final configuration was set to 16-bit watermark, 3 repetitions, alpha = 0.75, frequency profile = mid_high_quality and pair gap = 1.

5.1 Clean-Channel Imperceptibility Results

The first evaluation was performed on the visual quality of the stego images without any attack. The key imperceptibility findings are given in Table 5.

Table 5. Clean-channel imperceptibility results of the proposed method.

Metric	Result
Number of images	5,524
Average Peak Signal-to-Noise Ratio (PSNR)	47.386 dB
Minimum PSNR	30.745 dB
Maximum PSNR	60.105 dB
Average Structural Similarity Index Measure (SSIM)	0.995079
Minimum SSIM	0.944147
Maximum SSIM	0.999160
Average Mean Squared Error (MSE)	1.4461

The average PSNR value of 47.386 dB indicates that the proposed method causes very little distortion at the pixel level. In image hiding research, the higher PSNR value means that the stego image is close to the cover image. The average SSIM is 0.995079 which also indicates that the structure of the image is maintained after embedding. This is significant because SSIM is not just a measure of pixel error, but structural similarity. The high PSNR and high SSIM indicate that the proposed embedding process is visually light. The minimum PSNR is 30.745 dB, indicating that some images are more sensitive to embedding than others. This can occur if an image is smoother or less textured. The average result over 5,524 images is however good, indicating that the method is effective over a large and diverse set of images. The maximum PSNR of 60.105 dB shows that for many images the embedding distortion is extremely low.

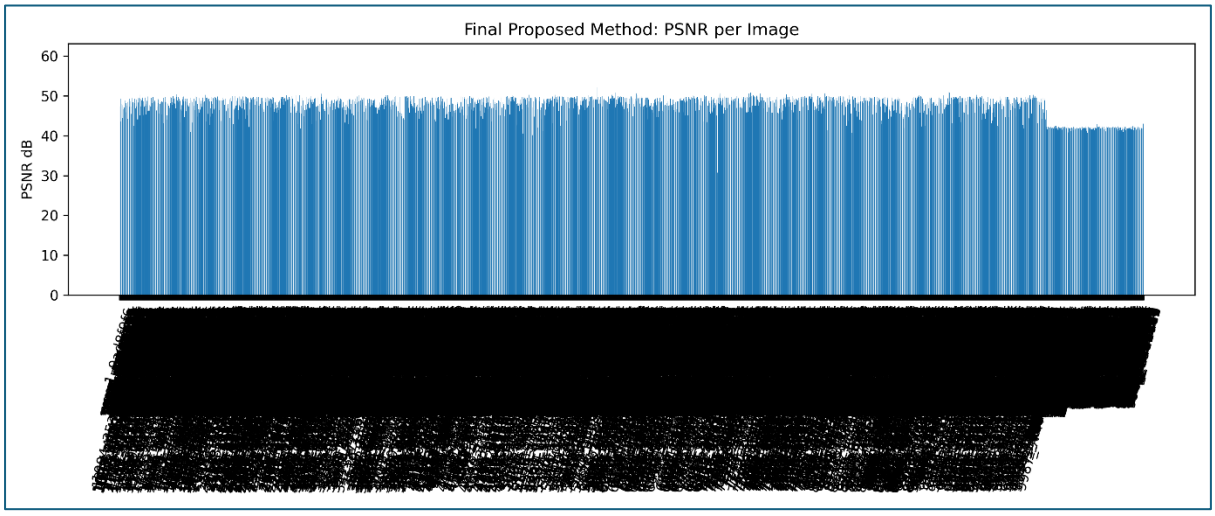


Figure 3. PSNR distribution of stego images

The PSNR values for the images on a per image basis are shown in figure 3 for the proposed adaptive LSB and lightweight Global-DCT framework. As seen from the figure, majority of the stego images are in the high PSNR range, primarily in the mid-40 dB to near 50 dB region. This indicates that the embedding process for the input images introduces only slight distortion at the pixel level for most of the images available in the dataset. The figure also confirms the average PSNR value obtained in Table 5. The average PSNR of the proposed method was 47.386 dB, showing high imperceptibility for 5,524 images. The highest PSNR obtained was 60.105 dB, indicating that the embedding modifications were very small for some images. Some of the images gave lower PSNR values, the lowest PSNR value obtained was 30.745 dB. This is a natural variation due to different images possessing different texture, smoothness, luminance distribution and edge density. Embedding changes are typically more readily apparent in the pixel-level measurements of images with smoother regions, and are more likely to be masked by small changes in textured images. As can be seen in Figure 3, the proposed method is stable over a large set of images and not just a few selected images. This result, in conjunction with the average SSIM of 0.995079 and low MSE of 1.4461 in Table 5, validates the proposed framework that maintains high visual quality while embedding the encrypted payload and compact copyright/authentication watermark..

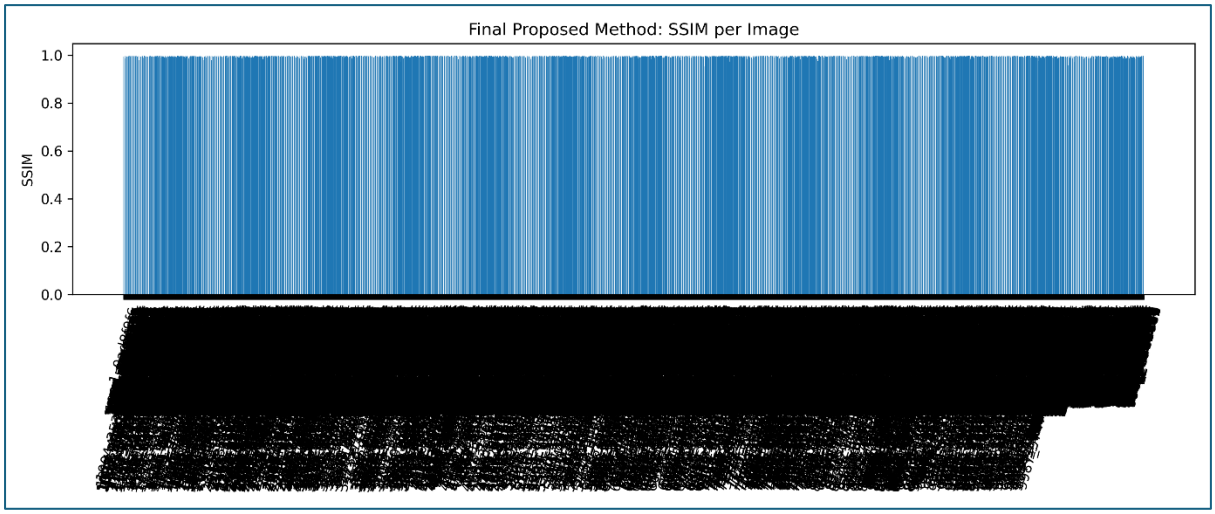


Figure 4. SSIM distribution of stego images.

The image-wise Structural Similarity Index Measure (SSIM) values of the proposed method is shown in Figure 4. The values are very close to 1.0 for nearly all the images, indicating that the structural content of the cover images is very well preserved after embedding. This is crucial as SSIM considers luminance, contrast and structural similarity, providing a more comprehensive understanding of visual quality perceived by the viewer than pixel error alone. This figure should be interpreted in conjunction with Figure 3 and Table 5. As can be seen in Figure 3, the method has a high average Peak Signal-to-Noise Ratio (PSNR) of 47.386 dB, and Figure 4 shows that the visual structure is also preserved, with an average SSIM of 0.995079. The lowest SSIM value is 0.944147, which is still reasonably high. The highest SSIM value is 0.999160, indicating that many stego images are almost structurally the same as their cover images. In general, the results of Figure 4 support the imperceptibility statement of the proposed framework. PSNR verifies that the distortion is low, and SSIM verifies that the embedding process does not noticeably affect the image structure. The PSNR and SSIM results indicate that the proposed adaptive LSB and lightweight Global-DCT embedding process maintains high visual quality for the tested image set of 5,524 images.

5.2 Payload and Watermark Recovery in Clean Condition

The results of the clean channel recovery are presented in Table 6. These results are used to check the correctness of the extraction of the encrypted payload and the watermark from the stego image without any attack.

Table 6. Clean-channel payload and watermark recovery results.

Metric	Result
Clean LSB payload extraction success	100%
Average adaptive LSB capacity	0.393 bpp
Average clean watermark Bit Error Rate (BER)	0.01668
Average clean watermark Normalized Correlation (NCC)	0.98596
Watermark length	16 bits
Watermark repetition	3

The clean LSB payload extraction success rate is 100%, meaning that the encrypted payload was extracted from all the images processed. This is an important result as the payload is not only hidden but also compressed, encrypted, extracted, decrypted, decompressed and verified using the Secure Hash Algorithm 256-bit (SHA-256) authentication hash. The clean extraction result shows that the adaptive LSB layer is reliable enough when it is not attacked. Average adaptive LSB capacity is 0.393 bpp, which indicates that the method can offer useful embedding space for payloads while maintaining good visual quality. The watermark extraction also works well in clean condition. The average clean watermark BER is 0.01668 and the NCC is 0.98596. The low BER and high NCC indicate that the embedded watermark in the compact Global-DCT is very similar to the original watermark. The watermark BER is not zero, as the watermark setting is very light. The selected watermark is only 16 bits with 3 repetitions, and the embedding strength is only 0.75. This light setting was deliberately chosen to maintain high PSNR and SSIM. Thus, the small clean watermark error is acceptable as the method emphasizes high imperceptibility while maintaining the watermark mostly recoverable.

5.3 Qualitative Visual Analysis

While numerical metrics are important, visual comparison is also required to understand whether the embedding changes are noticeable enough or not. Therefore, a qualitative comparison between cover images, stego images and amplified difference images should be included in the paper.

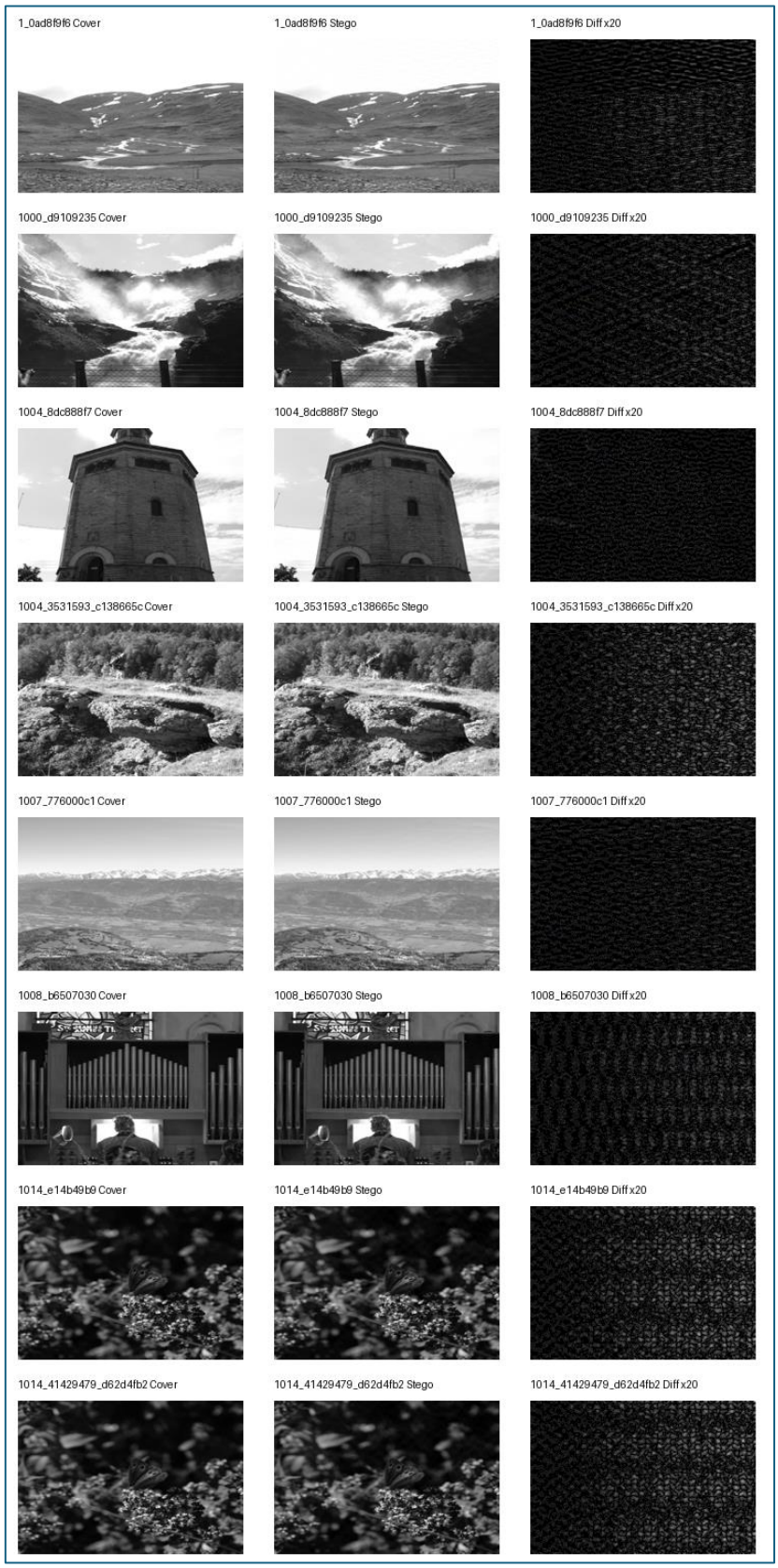


Figure 5. Visual comparison of cover images, stego images, and amplified difference images.

Sample visual results of the proposed method are shown in Fig. 5. Three images are shown in each row: the original cover image, the stego image generated and the amplified difference image. In normal viewing, the cover and stego images are almost visually indistinguishable. This demonstrates that the adaptive LSB and lightweight Global-DCT embedding process does not cause any noticeable visual artifacts. The difference image, amplified by a factor of 20, is shown in the third column. The view is exaggerated to expose the hidden changes for analysis. The difference maps indicate that the variations are small and spread throughout the image, not localized in a single area. This helps to design the proposed method in which the encrypted payload is embedded in the edge and texture sensitive area by adaptive LSB and the compact watermark is embedded by selected Global-DCT coefficient pairs.

The visual comparison also helps to substantiate the quantitative results presented in Table 5. The average PSNR of the stego images obtained by the proposed method is 47.386 dB and the average SSIM is 0.995079, which indicates that the stego images are very close to the original cover images. Thus, it can be concluded from the PSNR and SSIM values that the proposed framework has high imperceptibility when embedding the encrypted payload and the copyright/authentication watermark as shown in Figure 5. The difference images are also important, but they are amplified. They do not exhibit normal visual appearance rather they amplify the pixel changes caused by embedding. These difference images give an indication of the location of the embedding modifications. In the proposed method, the changes should be small and spread out, not large and noticeable. This helps to justify the design decision of embedding adaptive LSB in textured or edge-sensitive areas and embedding a light weight 16-bit Global-DCT watermark..

5.4 Robustness Against Image Processing Attacks

The robustness of the watermark was evaluated against JPEG compression, Gaussian noise, salt and pepper noise, resizing, blur, sharpening and cropping. The average watermark BER and NCC for each attack are shown in Table 7.

Table 7. Watermark robustness under image processing attacks.

Attack	Average BER	Average NCC	Interpretation
No attack	0.0167	0.9860	Strong clean extraction
Blur r1	0.0062	0.9949	Excellent
Resize 0.90	0.0273	0.9775	Strong
Resize 0.75	0.0288	0.9764	Strong
JPEG Q95	0.0314	0.9739	Strong
Gaussian σ2	0.0441	0.9633	Good
Gaussian σ3	0.0583	0.9509	Good
JPEG Q90	0.0612	0.9491	Good
Sharpen	0.0735	0.9398	Acceptable
JPEG Q80	0.1535	0.8707	Moderate
Salt-and-pepper 0.002	0.1581	0.8701	Moderate
JPEG Q70	0.2206	0.8111	Weak to moderate
JPEG Q60	0.2641	0.7726	Weak
Crop restore 0.90	0.4921	0.5636	Failed
Crop restore 0.95	0.4965	0.5566	Failed

Table 7 illustrates that the proposed lightweight watermark is very robust against blur, resizing, JPEG Q95, Gaussian noise and JPEG Q90. The best attack result is under blur with a BER of 0.0062 and an NCC of 0.9949. This means that the watermark is still very close to the original watermark after some blurring. The resize attacks also perform well with BER of 0.0273 for resize 0.90 and 0.0288 for resize 0.75. From these results, it can be seen that the Global-DCT coefficient-pair watermark is more resistant to resizing than the simple pixel domain

watermark. JPEG compression exhibits a progressive degradation. Under JPEG Q95, the BER is 0.0314, and under JPEG Q90 it becomes 0.0612. These values are still fine for a small and light watermark. But the higher the compression, the lower the performance. The BER rises to 0.1535 for JPEG Q80 and 0.2206 for JPEG Q70. The weakest compression result is JPEG Q60 (BER 0.2641). This indicates that the method is more appropriate for high quality and medium quality JPEG compression than for very heavy compression.

Noise attacks show mixed behavior. The Gaussian noise provides acceptable results, with BER 0.0441 for $\sigma = 2$ and 0.0583 for $\sigma = 3$. This implies that the watermark is fairly robust to the addition of small random noise. Salt-and-pepper noise is more challenging because it produces impulsive changes in the pixel values. With the salt-and-pepper noise, the BER rises to 0.1581, which is less than Gaussian noise, but better than cropping and heavy JPEG compression. Sharpening produces a BER of 0.0735 and NCC of 0.9398. This is fine, but it demonstrates that enhancement filters can interfere with the coefficient-pair relationship that is utilized in watermark extraction. The outcome is still valuable as the NCC is still high.

Sharpening produces a BER of 0.0735 and NCC of 0.9398. This is acceptable, but it shows that enhancement filters can disturb the coefficient-pair relationship used during watermark extraction. The result is still useful because the NCC remains high.

The weakest part of the method is the crop-and-restore attacks. Crop restore 0.90 gives BER 0.4921, and crop restore 0.95 gives BER 0.4965. These values are near to random extraction. This is because the image geometry is altered and the coefficient structure is shifted when cropping. No synchronization markers, feature-point registration, or geometric alignment are used in the proposed method hence, cropping should be considered as a weakness, not a strength.

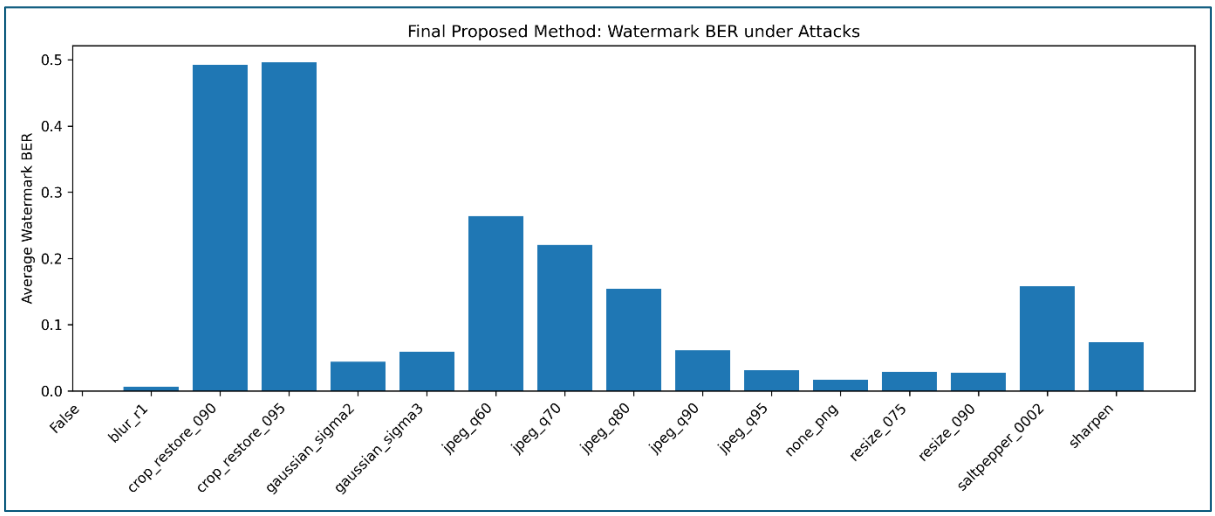


Figure 6. Watermark BER under different image processing attacks.

The average watermark Bit Error Rate (BER) for the different attacks tested is visually compared in figure 6. The lower the BER, the better the watermark recovery and the higher the BER, the more number of watermark bits were extracted incorrectly. As seen in the figure, the proposed method is robust for mild or common image processing operations. Specifically, blur r1, resize 0.75, resize 0.90, JPEG Q95, JPEG Q90, and Gaussian noise maintain the BER at a relatively low level. The clean condition (none_png) is also very strong, with a BER value close to 0, confirming reliable watermark extraction when no attack is applied. The figure also indicates a significant performance decline under higher distortions. It can be seen that JPEG Q80, JPEG Q70 and particularly JPEG Q60 result in significantly higher BER values, indicating that the relationships between the selected Discrete Cosine Transform (DCT) coefficients are more disturbed by stronger compression. This can also be observed under salt-and-pepper noise, where the BER increases compared to Gaussian noise,

because impulse noise causes larger local pixel perturbations.

The most significant observation in Figure 6 is the dramatic increase under crop_restore_090 and crop_restore_095, where the BER is close to 0.49–0.50. This is near random recovery and indicates that cropping is the most challenging attack for the proposed method. Cropping is not compression or mild filtering, but it alters the spatial/frequency alignment of the image. Therefore, the original Global-DCT coefficient-pair structure is not well preserved to be extracted reliably. This behavior indicates that a different synchronization method, such as template-based alignment, feature-point registration, or another geometric correction method is necessary for geometric attacks.

The overall conclusion from Figure 6 is that it is consistent with the robustness analysis reported in Table 7. The proposed framework is robust against blur, resizing, high quality JPEG compression and Gaussian noise, while stronger JPEG compression, impulse noise and particularly cropping make watermark recovery more difficult.

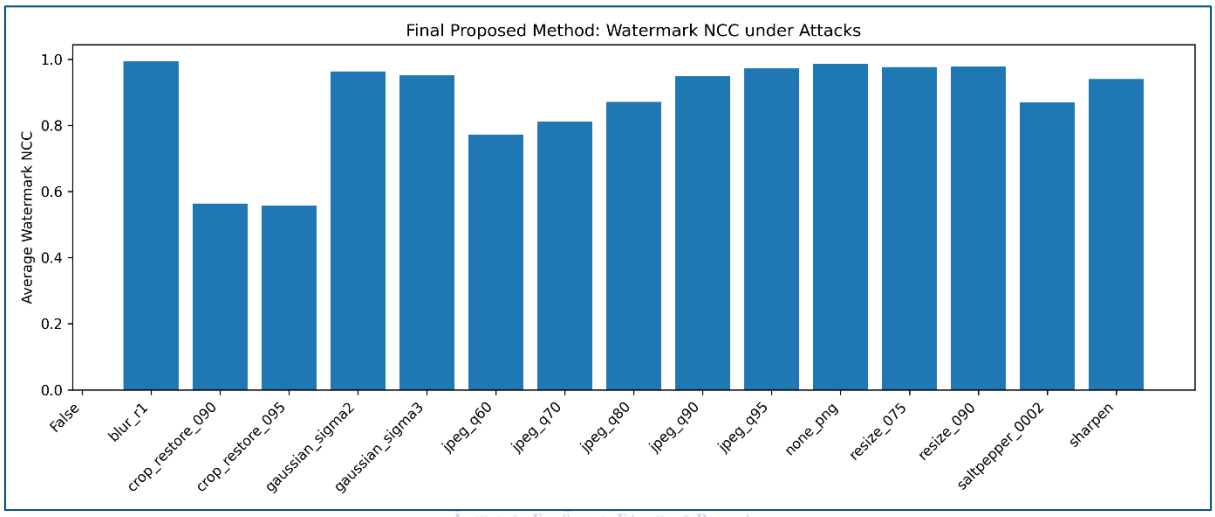


Figure 7. Watermark NCC under different image processing attacks.

Figure 7 is to be read in conjunction with Figure 6 as it represents watermark recovery from a different perspective. Figure 6 is the Bit Error Rate (BER) and Figure 7 is the Normalized Correlation (NCC) between the original watermark and the extracted watermark. A lower BER and a higher NCC both represent better watermark recovery, in general. It can be seen from the figure that the proposed method has high NCC values under the blur r1, resize 0.75, resize 0.90, JPEG Q95, JPEG Q90 and Gaussian noise. In these cases, the NCC values are still close to 1.0, that is, the extracted watermark is still very similar to the original watermark. This is in complete agreement with the results shown in Figure 6, in which the same attacks resulted in relatively low BER values. The clean condition (none_png) also exhibits high NCC, indicating that the extraction is reliable without the application of attack.

Under higher compression settings, a gradual decrease can be observed. For instance, JPEG Q80, JPEG Q70 and particularly JPEG Q60 have a more noticeable effect on NCC. This implies that the more compression, the more the watermark structure is disrupted and the less similar the watermark is to the original watermark. A similar decrease can be observed under salt-and-pepper noise, which is more detrimental than Gaussian noise, since it introduces sudden disturbances at the pixel level. The lowest NCC values are found under crop_restore_090 and crop_restore_095, with NCC reaching almost 0.56. This outcome is consistent with the high BER values that have been seen in Figure 6 and validates that the most harmful attack on the proposed method is cropping. Cropping changes the image overall alignment and hence destroys the original coefficient structure, so that the watermark in

the image cannot be recovered as reliably as it can normally under non-geometric attacks.

Hence, the overall performance of Figures 6 and 7 indicates that the proposed framework is resilient to blur, resizing, high-quality JPEG compression and Gaussian noise, and more severe compression and geometric distortion have more negative effects on watermark similarity and recovery.

5.5 Image Quality After Attacks

The attack evaluation also assessed the quality of the images following each attack. This is helpful as some attacks seriously warp the image itself. The greater the change in the image caused by the attack, the more difficult it is to recover the watermark.

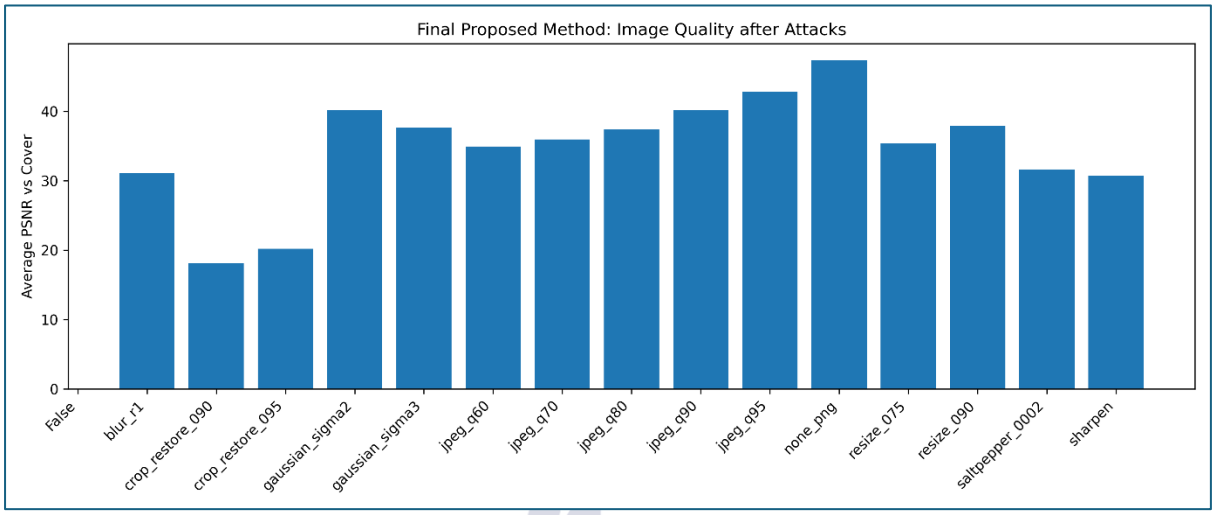


Figure 8. Image quality after attacks measured using PSNR.

The average PSNR of the original cover images and attacked stego images is shown in figure 8. This number can be used to understand why certain attacks degrade the watermark recovery more than others. Generally, if the attacked image is still visually and numerically similar to the original cover image, the embedded watermark is more likely to be recoverable. The more intense the visual or structural changes due to the attack, the more difficult it is to recover the watermark. As can be seen from the figure, the PSNR is the highest for the none_png condition, which is understandable since no attack is performed. JPEG compression also retains image quality well, if the compression quality is high. For instance, JPEG Q95 and JPEG Q90 have higher PSNR than higher compression rates. The PSNR decreases from JPEG Q95 to JPEG Q80, JPEG Q70 and JPEG Q60 as compression increases. This trend is consistent with the BER trend in Figure 6, which shows that the higher the JPEG compression, the higher the watermark error.

Moderate to high PSNR values are observed for the resize attacks. This is in line with the previous watermark results, which indicate that watermark resize 0.90 and watermark resize 0.75 achieve relatively low BER and high NCC. This implies that the values of the pixels will be altered when resizing, but the proposed Global-DCT coefficient-pair watermark is still relatively stable after resize-and-restore operations. The degradation pattern of the Gaussian noise is also controlled. As expected, Gaussian σ2 maintains better image quality than Gaussian σ3. This is consistent with the watermark results, which showed that Gaussian σ3 had slightly higher BER than Gaussian σ2. The PSNR of the crop-and-restore attacks are significantly lower than most other attacks. This is a confirmation that cropping does not just alter the value of the pixels, but also affects the geometry of the image. This geometric disturbance is the reason why the BER of crop_restore_090 and crop_restore_095 is near the random extraction in Figure 6.

Overall, Fig. 8 relates the image degradation to the watermark recovery. The watermark recovery of attacks that preserve higher PSNR, like none_png, JPEG Q95/Q90, resizing, blur and moderate Gaussian noise, is generally better.

Strong attacks (such as heavy JPEG compression and cropping) that affect PSNR or affect geometry more strongly yield poorer watermark extraction results.

5.6 Overall Discussion

The results indicate that the proposed method is able to achieve a good balance between imperceptibility and lightweight watermark robustness. The primary advantage of the technique is the quality of the images in the clean channel. The average PSNR value of the stego images is 47.386 dB and the average SSIM value is 0.995079, which indicates that the stego images are very similar to the original images. The method also obtains 100% clean LSB payload extraction, which verifies that the encrypted payload can be recovered correctly before attack. The Global-DCT watermark is light weight and offers some useful robustness against a number of common operations. It is particularly effective under blur, resizing, JPEG Q95/Q90 compression and Gaussian noise. This is useful because these operations are frequently performed when images are manipulated, stored, shared, resized, or mildly modified.

The results also demonstrate the compromise of the lightweight design. Only a 16-bit watermark with 3 repetitions and alpha 0.75 is used in the method. This ensures high image quality, but it also reduces the robustness against stronger attacks like JPEG Q80/Q70, JPEG Q60, salt-and-pepper noise, and cropping. Increasing the strength of the watermark would lower BER for these attacks, but would also lower PSNR and SSIM. Thus, the proposed method is not a fully attack-invariant watermarking method, but a high fidelity image hiding method with useful watermark robustness. The proposed method is simpler and more reproducible than heavy hybrid watermarking methods and deep learning-based watermarking methods [8], [9], [13], [14], [16]. It does not need training data or optimization of the neural network. Meanwhile, it enhances the LSB-only hiding with a transform domain copyright/authentication watermark. This is ideal for situations where image quality, payload concealment, authentication and ease of implementation are all critical. The primary constraint is crop robustness and strength. Cropping alters the geometry of the image and the existing Global-DCT coefficient selection process does not involve alignment recovery. This can be enhanced in the future by incorporating synchronization templates, feature-point matching, or redundant embedding of the watermark in regions.

6. CONCLUSION AND FUTURE ENHANCEMENTS

In this study, Adaptive Least Significant Bit (LSB) and Lightweight Global Discrete Cosine Transform (Global-DCT) Framework for High-Fidelity Image Steganography and Robust Copyright Authentication was proposed. The technique integrates the encrypted payload hiding with compact watermark embedding in a single image hiding framework. The adaptive LSB layer conceals the payload (encrypted data) into the textured and edge-sensitive image areas, while the Global-DCT layer carries a small copyright/authentication watermark by modifying the coefficient pairs.

The proposed method was tested on 5,524 images obtained from Kodak, Uncompressed Color Image Database (UCID) and BOSSBase/BOWS2 image sources. The final implementation was based on a 16 bit watermark, 3 repetitions, alpha = 0.75, the mid_high_quality frequency profile, and pair gap = 1. These parameters were chosen to ensure that the watermark is light, yet still maintains high visual quality. The clean-channel results demonstrate good imperceptibility and payload recovery of the proposed method. It obtained an average Peak Signal-to-Noise Ratio (PSNR) of 47.386 dB, average Structural Similarity Index Measure (SSIM) of 99.5079% and average Mean Squared Error (MSE) of 1.4461. The encrypted payload was successfully recovered by 100% clean LSB extraction and the average adaptive LSB capacity was 0.393 bits per pixel. The compact watermark also exhibited good clean extraction performance with an average Bit Error Rate (BER) of 1.668% and average Normalized Correlation (NCC) of 98.596%.

The robustness results indicate that the light weight Global-DCT watermark is still useful under a number of common image processing operations. The method was found to be performing well under blur, resizing, JPEG Q95/Q90 compression and Gaussian noise.

These results show that the proposed coefficient-pair watermark is capable of copyright/authentication verification following common image sharing and editing processes. The compression strength, impulse noise, and geometric changes create more challenging conditions which is expected because the applied method intentionally uses a lightweight watermark to preserve and maintain the quality of an image.

Future enhancements can focus to extend the proposed framework in a number of ways in several directions. First, synchronization-assisted extraction can be added to enhance the performance under cropping, rotation and other geometric transformations. Second, more powerful error-correction coding can be used for the watermark bits to enhance recovery in the presence of more aggressive compression and impulse noise. Third, embedding strength can be adaptive, based on local texture, luminance and frequency energy. Fourth, the method can be tested with the current steganalysis detectors to assess the statistical detectability, as well as the visual quality and watermark recovery [20, 26]. Finally, the framework can be tested in the future on domain-specific datasets like medical images, satellite images, document images and social-media-compressed images.

Overall, the proposed framework method offers a simple, lightweight, and reproducible solution for image hiding. It not only integrates encrypted payload hiding and copyright/authentication watermarking but also avoids the training cost of deep learning-based methods. The results suggests that the method is suitable for applications where high visual-quality, clean payload recovery, and practical robustness against common image processing operations are desired.

REFERENCES

[1] Song, B., Li, S., Xu, X., and Qin, J. 2024. "A survey on deep-learning-based image steganography." *Expert Systems with Applications*, 245: 123090.

[2] Aljughaiman, A., Alrawashdeh, R., and Almuhammadi, S. 2025. "Content-adaptive LSB steganography with saliency fusion, ACO dispersion, and hybrid encryption with ablation study." *Scientific Reports*, 15: Article 33920.

[3] Sanjalawe, Y., Al-E'mari, S., Fraihat, S., Abualhaj, M., and Alzubi, E. 2025. "A deep learning-driven multi-layered steganographic approach for enhanced data security." *Scientific Reports*, 15: Article 4761.

[4] Alrawashdeh, R., Almuhammadi, S., and Niazi, M. 2025. "Secure edge-guided adaptive image steganography using HED-based attention maps and CNN." *Scientific Reports*, 15: Article 27150.

[5] Ismail, S. M., et al. 2026. "Edge-adaptive high-capacity image steganography using hybrid edge detection and MSB embedding." *Computers*, 15(3): 141.

[6] Ji, P., et al. 2025. "Edge-guided dual-stream U-Net for secure image steganography." *Applied Sciences*, 15(8): 4413.

[7] Alanzy, M. 2023. "Image steganography using LSB and hybrid encryption algorithms." *Applied Sciences*, 13(21): 11771.

[8] Hosny, K. M., Darwish, M. M., and Li, K. 2024. "Digital image watermarking using deep learning: A survey." *Computer Science Review*, 53: 100662.

[9] Ben Jabra, S., Koubaa, A., and Ammar, A. 2024. "Deep learning-based watermarking techniques: Challenges and opportunities." *Circuits, Systems, and Signal Processing*, 43: 1–31.

[10] Luo, Y., et al. 2024. "Robust deep image watermarking: A survey." *Computers, Materials & Continua*, 81(1): 1–33.

[11] Bistroń, M., et al. 2026. "Deep learning for image watermarking: A comprehensive survey." *Sensors*, 26(2): 444.

[12] Dong, Y., Wang, J., and Li, X. 2024. "An adaptive robust watermarking scheme based on chaotic mapping and hybrid transform domain." *Scientific Reports*, 14: Article 76101.

[13] Chaudhary, H., Garg, P., and Vishwakarma, V. P. 2025. "Enhanced medical image watermarking using hybrid DWT-HMD-SVD and Arnold scrambling." *Scientific Reports*, 15: Article 94080.

[14] Alrammahi, A., et al. 2025. "A robust image watermarking based on DWT and RDWT combined with Mobius transformations." *Computers, Materials & Continua*, 84(1): 1–20.

[15] Hebbache, K., et al. 2024. "A DWT-based approach with gradient analysis for robust medical image watermarking." *Applied Sciences*, 14(14): 6199.

[16] Shubuh, S., et al. 2024. "Robust image watermarking based on hybrid IWT-DCT-SVD." *International Journal of Computer Science and Mobile Computing*, 13: 1–12.

[17] Kouadri, A., et al. 2025. "A robust blind hybrid watermarking technique for medical image security." *EURASIP Journal on Advances in Signal Processing*, 2025: Article 1275.

[18] Gharib, H. A., et al. 2025. "Robust zero-watermarking for color images using hybrid feature extraction." *Scientific Reports*, 15: Article 09290.

[19] Wei, S., et al. 2025. "Universal image vaccine against steganography." *Symmetry*, 17(1): 66.

[20] Denemark, T., Sedighi, V., Holub, V., Cogranne, R., and Fridrich, J. 2014. "Selection-channel-aware rich model for steganalysis of digital images." *IEEE International Workshop on Information Forensics and Security*, 48–53.

[21] Bas, P., Filler, T., and Pevný, T. 2011. "Break our steganographic system: The ins and outs of organizing BOSS." *Information Hiding, Lecture Notes in Computer Science*, 6958: 59–70.

[22] Franzen, R. 2013. "Kodak Lossless True Color Image Suite." Available online: Kodak image suite archive.

[23] Schaefer, G., and Stich, M. 2004. "UCID: An uncompressed color image database." *Proceedings of SPIE*, 5307: 472–480.

[24] Binghamton University Digital Data Embedding Laboratory. 2011. "BOSSBase 1.01 image database." Binghamton University, Department of Electrical and Computer Engineering.

[25] Gupta, A. 2023. "BOWS2." *Mendeley Data*, Version 1. DOI: 10.17632/kb3ngxfmjw.1.

[26] Huang, S., et al. 2024. "Forensics aided content selection network for steganalysis." *Pattern Recognition*, 2024: 1–15.

[27] Zhang, S., et al. 2025. "A multi-image steganography scheme using image source selection." *Journal of Cloud Computing*, 14: Article 3336.