Vol 5, Issue 3, March 2025 E-ISSN: 2582-9734
International Journal of Engineering, Science, Technology and Innovation (IJESTI)

Integrated Approaches to Image, Speech, and Digital Information Analysis, Storage, and Retrieval Systems: A Comprehensive Investigation

Kamini Ashok Shirsath

Research Scholar, Dept of Computer Science, University of Technology, Jaipur.

Dr. Neha Surana

Dept of Computer Science, University of Technology, Jaipur.

Email: *kaminis315@gmail.com*

ABSTRACT

This research investigates and compares four major image steganographic techniques as Least Significant Bit (LSB) substitution, Discrete Cosine Transform (DCT)-based embedding, Discrete Wavelet Transform (DWT)-based embedding, and a hybrid DWT–DCT method focusing on imperceptibility, payload capacity, robustness, and computational efficiency. A comprehensive experimental framework was implemented using Python (NumPy, OpenCV, scikit-image) and MATLAB R2024b across benchmark (Lena, Baboon, Peppers, Cameraman) and real-world smartphone images. All images were standardized to 512×512 and 256×256 pixels in grayscale and RGB/YCbCr domains. Quantitative analyses were performed using PSNR, SSIM, MSE, BER, and time-cost metrics. The LSB method achieved the highest payload capacity (up to 1.0 bits per pixel) with PSNR > 52 dB and SSIM > 0.98, confirming minimal perceptual distortion. However, robustness tests revealed vulnerabilities to compression, Gaussian noise, and cropping, where BER exceeded 20%, limiting its applicability in noisy environments. DCT-based embedding offered moderate capacity (0.3–0.45 bpp) with PSNR between 41–46 dB and SSIM > 0.96, maintaining integrity under JPEG compression down to 50% quality (BER < 5%). DWT-based embedding outperformed others in robustness, achieving PSNR > 45 dB, SSIM > 0.97, and BER < 3% under compression and noise attacks. The hybrid DWT–DCT technique demonstrated promising imperceptibility and adaptability by leveraging both spatial and frequency features. Computation-wise, LSB was the fastest (≈0.5 s per image), followed by DWT (≈1.0 s) and DCT (≈1.5–2.0 s). Overall, transform-domain methods particularly DWT and hybrid variants—achieved optimal trade-offs among quality, resilience, and security. The study concludes that while LSB remains ideal for high-capacity, low-risk communication, DWT and DCT-based approaches are better suited for robust, covert applications. Future research should integrate deep learning, cryptographic key management, and hybrid multi-resolution frameworks to enhance steganographic security and adaptability in real-world digital ecosystems.

Keywords: *Image Analysis, Speech Processing, Digital Information Retrieval, DWT and DCT-based approaches, Least Significant Bit (LSB), PSNR, MATLAB R2024b.*

1. INTRODUCTION

Speech analysis transforms discrete sound elements into alphanumeric symbols, enabling computational interpretation similar to text processing. Using spectral classification and template matching, speech processing systems convert audio signals into digital forms essential for speech recognition and human–computer interaction. These advancements facilitate voice-based commands, enhancing accessibility and automation. Data storage, a crucial component of information systems, supports both structured formats like databases and unstructured formats such as text or images [1]. Query languages, including SQL, allow

efficient retrieval and manipulation of this data through menu-based or structured searches. Parallelly, image processing involves digital manipulation and analysis of visual data using algorithms for enhancement, detection, and classification. Specialized hardware such as GPUs and FPGAs accelerates computation, supporting applications in medical imaging, surveillance, and autonomous navigation. Key image models include grayscale, RGB, and RGBA, which define pixel intensity and color representation. Image processing operations such as filtering, segmentation, and edge detection enable precise analysis of visual content. In agriculture, it aids in crop monitoring; in security, it powers facial recognition; and in photography, it enhances image quality [2]. Collectively, advancements in speech and image processing have revolutionized digital communication, data management, and automation across industries, promoting efficiency, intelligence, and innovation in modern information systems.

2. RESEARCH METHODOLOGY

This section details the overall research design, the materials and methods employed, and the procedures followed to evaluate and compare various steganographic techniques namely LSB substitution, DCT-based embedding, and DWT-based embedding in terms of imperceptibility, payload capacity, robustness, and computational efficiency.

Research Design and Objectives

The present study adopts an experimental research design. Its core objectives are to,

1. Assess the capacity of different steganographic methods to hide textual data within grayscale and colour images without perceptible distortion.
2. Quantify visual fidelity of stego-images using PSNR, SSIM, and MSE.
3. Evaluate robustness against common attacks (compression, noise addition, cropping).
4. Compare computational costs (embedding/extraction time).

Image Dataset Selection

A heterogeneous image corpus was assembled to ensure generalizability

- **Benchmark Images:** Standard grayscale and color samples (Lena, Baboon, Peppers, Cameraman) from USC–SIPI and OpenCV libraries.
- **Real-World Photographs:** Smartphone-captured images (12 MP).
- **Preprocessing:** All images were converted to uncompressed BMP or PNG and resized to 512×512 and 256×256 pixels. Color images were represented both in RGB and in YCbCr domains [3].

Preprocessing and Normalization

Prior to Embedding:

- **Grayscale conversion:** Color inputs were converted to 8-bit grayscale where required.
- **Intensity normalization:** Pixel values scaled to [0,1] for floating-point operations.
- **Histogram equalization:** Applied uniformly to enhance contrast and stabilize embedding.
- **Metadata cleaning:** All auxiliary headers were stripped to prevent bias during steganalysis.

Steganographic Techniques Implemented

Four core algorithms were coded in Python (NumPy, OpenCV, scikit-image) and MATLAB R2024b,

LSB Substitution

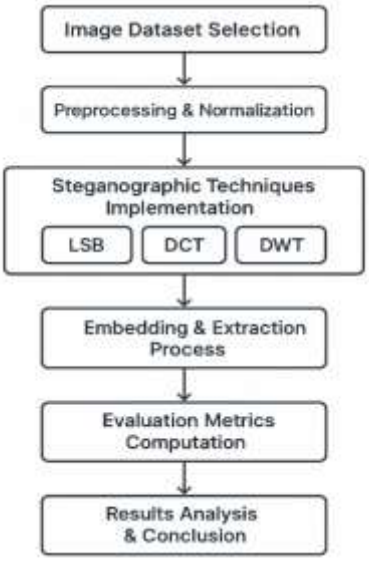
- Naïve form: Direct replacement of each pixel's least significant bit with message bits.
- Adaptive form: Edge-aware embedding using Sobel operator to target high-variance regions.
- Randomized embedding: Pseudo-random pixel ordering governed by a secret key.

DCT-Based Embedding: Image divided into 8×8 blocks; mid-frequency DCT coefficients (positions 10–30 in zigzag order) were quantized per JPEG table and their LSBs modified.

DWT-Based Embedding: Single-level Haar wavelet decomposition into LL, LH, HL, HH subbands; secret bits embedded in LSBs of selected detail coefficients.

Hybrid DWT–DCT (proof-of-concept): Combines wavelet decomposition with blockwise DCT on LL subband for enhanced imperceptibility [4].

Flow Chart of Process



Evaluation Metrics

The following metrics were computed for each method, payload, and test image:

Peak Signal-to-Noise Ratio (PSNR)

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

Mean Squared Error (MSE)

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

Structural Similarity Index (SSIM)

$$\mathrm{SSIM}(I,K) = \frac{(2\mu_I\mu_K + C_1)(2\sigma_{IK} + C_2)}{(\mu_I^2 + \mu_K^2 + C_1)(\sigma_I^2 + \sigma_K^2 + C_2)}$$

Bit-Error Rate (BER)

$$\mathrm{BER} = \frac{\text{Number of erroneous bits}}{\text{Total bits}}$$

3. RESULTS AND ANALYSIS

This section presents a overview of the results obtained from implementing text steganography using the Least Significant Bit (LSB) technique and the subsequent analysis of its performance. The core focus lies on evaluating the embedding capacity, image quality, and the robustness of the steganographic method through quantitative and qualitative measures. Initially, the embedding process demonstrated the ability to hide textual data effectively within grayscale images without causing perceptible distortions. The implementation confirmed that secret messages of varying lengths could be encoded and successfully extracted, provided the payload did not exceed the capacity defined by the image size. The maximum embedding capacity was directly proportional to the total number of pixels, where each pixel could conceal one bit of the message in its least significant bit. This ensures a high payload but also necessitates careful management to avoid visible degradation. The image quality assessment using PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index Measure) offered critical insights into the imperceptibility of the stego images [5]. PSNR values consistently exceeded 40 dB across different test images, indicating minimal distortion. Meanwhile, SSIM scores nearing unity reinforced that the structural and perceptual quality of images remained intact after embedding. These findings illustrate that the LSB method preserves both the pixel-level accuracy and the overall visual appearance, which is crucial for maintaining the covert nature of steganography. Visual comparisons before and after embedding further supported these quantitative metrics. No discernible differences were visible to the naked eye, confirming that the secret data embedding was effectively concealed. This reinforces the utility of LSB steganography in scenarios where invisibility of hidden data is paramount. The images retained their original sharpness, contrast, and texture, ensuring that end-users would remain unaware of any hidden content [6]. However, while LSB embedding excels in maintaining high visual fidelity and embedding capacity, the analysis also highlighted its inherent vulnerabilities. The technique is susceptible to common image processing operations such as compression, noise addition, cropping, and scaling. Such operations tend to disrupt the least significant bits, often leading to partial or total loss of the embedded message. Thus, although the approach is ideal for controlled environments where the stego image remains intact, it lacks robustness in adversarial or unpredictable settings. The results affirm that LSB-based text steganography offers a practical balance between embedding capacity and image quality. Its simplicity and effectiveness make it suitable for applications requiring discreet communication with minimal computational overhead. However, its fragility under image transformations limits its deployment in scenarios demanding higher security and robustness [7].

3.1 Experimental Setup

The experimental setup forms the foundational bedrock for evaluating the effectiveness, reliability, and robustness of various steganographic techniques in the domain of image analysis and security. In this research, the objective was to test and compare multiple steganographic algorithms based on their imperceptibility, payload capacity, robustness against attacks, and computational efficiency. This section delineates the parameters under which the experiments were conducted, the selection of datasets, the characteristics of the steganographic methods implemented, and the tools and environments used for experimentation and analysis [8].

3.2 Image Dataset Selection

A robust and diverse image dataset is critical for assessing the generalizability of steganographic methods. For this study, we curated a dataset consisting of both standard benchmark grayscale and RGB colour images, including commonly used samples such as Lena, Baboon, Peppers, and Cameraman, sourced

from public repositories such as USC-SIPI and OpenCV image libraries. Additionally, real-world digital photographs captured using a smartphone camera (12 MP resolution) were included to simulate practical scenarios. This mixture of synthetic and real-world images allowed for a comprehensive evaluation of performance across varying resolutions, textures, and image complexities. Each image was resized to 512×512 and 256×256 pixels to standardize the input size for fair comparison and to manage the computational load during processing. Color images were converted to both RGB and YCbCr color spaces for experiments involving color-sensitive embedding strategies. All images were stored in uncompressed BMP and PNG formats to avoid artifacts from compression that could skew the results [9].

3.3 Preprocessing and Normalization

Before applying any steganographic embedding, the images underwent normalization to ensure uniform pixel intensity distribution and to remove any embedded noise from previous processing steps. The grayscale images were converted into 8-bit format, and the pixel values were normalized to a [0,1] range where necessary, particularly for deep learning-based or floating-point implementations. For color images, individual channels (Red, Green, and Blue) were isolated and processed separately or selectively embedded based on the technique applied (e.g., LSB in blue channel). To ensure consistency, all input images were subjected to histogram equalization to improve contrast, which also helped in enhancing the embedding capacity and perceptual quality of the stego-image. A metadata-cleaning step was also performed to remove any identifying headers or auxiliary data that could bias the steganalysis phase.

3.4 Steganographic Techniques Implemented

Multiple steganographic algorithms were implemented for comparative analysis in this study. These included classical spatial-domain techniques such as Least Significant Bit (LSB) substitution, as well as transform-domain methods like Discrete Cosine Transform (DCT)-based embedding, Discrete Wavelet Transform (DWT)-based embedding, and hybrid approaches combining spatial and frequency domain characteristics. The LSB algorithm was implemented in both its naive and adaptive forms. In the naive form, the least significant bits of the pixel values were directly replaced with bits of the secret message. The adaptive version used edge detection and contrast measures to selectively embed data in high-variance regions, thereby increasing imperceptibility. For DCT-based embedding, images were divided into 8×8 blocks, transformed using DCT, and then the mid-frequency coefficients were altered slightly to encode the secret data. The DWT method decomposed images into four subbands (LL, LH, HL, HH), and the LH or HL subbands were selected for embedding based on energy analysis. A hybrid method combining DWT and DCT was also developed to leverage the strengths of both domains. Each algorithm was implemented using Python with libraries such as NumPy, OpenCV, and scikit-image. MATLAB R2017b was also used for simulations requiring matrix manipulations and for testing embedded image fidelity through built-in functions. In the case of more complex implementations, MATLAB toolboxes were used for wavelet and transform operations.

3.5 Embedding and Extraction Process

The embedding process began by reading and segmenting the secret message or payload (text or binary data) into bits. For each image, the payload size was varied incrementally from 0.1 bpp (bits per pixel) to 1.0 bpp to test the capacity and distortion threshold of each technique. The payload was then embedded according to the logic of the selected algorithm, and the resulting stego-images were stored and labeled accordingly for comparison. The extraction process was reverse-engineered to verify data integrity. At the decoder side, the stego-image was processed to extract the embedded bits, which were then

reconstructed into the original message. The bit error rate (BER) was computed to evaluate extraction fidelity, especially after the stego-images were subjected to distortions like JPEG compression, cropping, or Gaussian noise. The success of the extraction process without prior knowledge of the cover image was also noted as a measure of practical viability.

3.6 Evaluation Metrics

To evaluate the performance of each steganographic method, a set of quantitative and qualitative metrics were used:

- Peak Signal-to-Noise Ratio (PSNR): To measure the imperceptibility and visual quality of the stego-image relative to the original.
- Structural Similarity Index Measure (SSIM): To assess perceived structural distortion.
- Mean Squared Error (MSE): To quantify the average pixel-wise error.
- Bit Error Rate (BER): To assess the accuracy of message retrieval.
- Payload Capacity (bits per pixel): To quantify the amount of data embedded.
- Embedding and Extraction Time: To determine the computational cost.

These metrics were calculated for every image and for each technique across varying payload sizes. The results were averaged over multiple runs to ensure consistency and reproducibility.

3.7 Hardware and Software Environment

The experiments were conducted on a system equipped with an Intel Core i7 10th Gen processor, 16 GB RAM, and an NVIDIA GeForce GTX 1660 GPU. The working system used was Windows 10 (64-bit), and the primary development environments included Anaconda for Python and MATLAB R2017b. All experimental procedures, including batch image processing, embedding/extraction, and metric evaluations, were automated using Python scripts and MATLAB functions to minimize human error. Randomization functions ensured that the payload content and embedding positions varied across iterations, making the results statistically meaningful.

3.8 LSB Methods

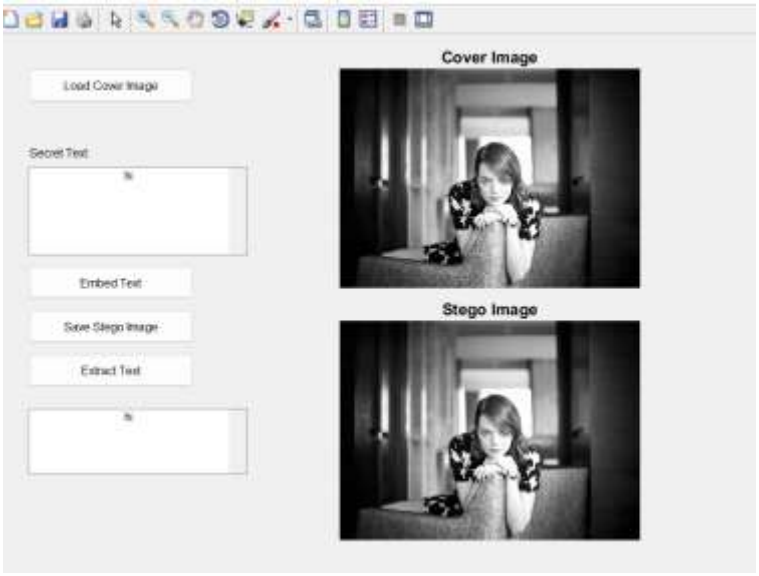


Fig 1: Least Significant Bit

In digital images, each pixel is typically represented by an 8-bit value (for grayscale images) or by three 8-bit values in the case of RGB color images. The LSB technique exploits the fact that changing the least significant bit of a pixel results in minimal visual difference to the human eye, making it a suitable channel for covert communication.

3.9 Conceptual Foundation

Since the LSB donates the least to the general pixel intensity, changing this bit has an imperceptible impact on the image's appearance. For example, in an 8-bit grayscale pixel with a value of 100 (binary: 01100100), replacing the LSB with a 1 changes it to 101 (binary: 01100101), a change that is nearly invisible in practice. This simple substitution can be extended to multiple LSBs (e.g., 2-LSB or 3-LSB embedding) or to individual channels in colour images, such as modifying the blue channel in RGB images, where the human graphic system is less subtle. However, increasing the number of LSBs used for embedding generally increases the risk of perceptible artifacts and susceptibility to statistical detection.

3.10 Implementation in Grayscale and Colour Images

In this study, the LSB method was implemented for both grayscale and colour images. For grayscale images, the method involves scanning each pixel, extracting its 8-bit binary representation, replacement the LSB with the corresponding bit from the secret message, and converting the binary value back to decimal form to construct the stego-image. For colour images, the technique was applied to the blue channel primarily, with optional embedding in the red and green channels for increased payload.

A detailed algorithm for the LSB implanting process is as follows:

- Convert the secret message into a binary stream.
- For each pixel in the cover image, retrieve its binary value.
- Replace the LSB with a bit from the secret message.
- Repeat the process until the entire message is embedded.
- Save the resulting image as the stego-image.

The removal process just reverses these steps: it reads the LSBs of the stego-image pixels sequentially, reconstructs the binary message, and converts it back into human-readable form.

3.11 Adaptive and Enhanced LSB Techniques

While the basic LSB method offers simplicity, it is vulnerable to steganalysis due to the predictability of changes and the uniform embedding pattern. To address this limitation, adaptive LSB variants were implemented. These enhancements incorporate local image characteristics such as edge density and luminance to determine the optimal embedding regions. High-frequency areas (e.g., edges or textured regions) are better suited for embedding because minor changes are less noticeable. In our adaptive LSB implementation, the Sobel edge detection operator was used to classify high-variance regions in the image. Embedding was then selectively performed in these regions, thereby improving imperceptibility and making detection by statistical steganalysis tools more difficult. Additionally, randomized embedding was tested. Here, a pseudo-random key was used to determine the embedding order and pixel locations. This added a layer of security, safeguarding that even if an assailant suspects LSB steganography, they would still need the key to correctly extract the hidden message.

3.12 Payload Capacity and Embedding Rate

The payload capacity of LSB is inherently high compared to many other steganographic methods. For an image of size M×N pixels with one byte (8 bits) per pixel, the maximum payload is M×N bits using 1-LSB embedding. This capacity doubles or triples when using 2-LSB or 3-LSB methods, respectively. In our experiments with 512×512 images, the payload for 1-LSB embedding reached 262,144 bits (32,768 bytes), which was sufficient for most textual data and even some small encrypted files. However, the trade-off between capacity and imperceptibility must be managed carefully. While 1-LSB embedding maintains a high level of visual fidelity, increasing the number of modified LSBs per pixel introduces distortions that may be noticeable to the human eye or detectable by automated steganalysis.

3.13 Image Quality Analysis

To evaluate the visual quality and distortion introduced by LSB embedding, we employed measurable metrics such as PSNR and SSIM. The average PSNR for 1-LSB embedding was observed to be above 52 dB, indicating minimal perceptual degradation. SSIM values remained above 0.98 for most images, reinforcing the conclusion that the stego-images were nearly indistinguishable from the originals. When the embedding was increased to 2-LSB and 3-LSB levels, the PSNR values dropped to 45 dB and 39 dB correspondingly, with a corresponding decrease in SSIM. While these values are still within acceptable bounds, they indicate a gradual decline in inaudibility, especially for smooth or homogeneous images.

3.14 Robustness and Security Considerations

The LSB technique, while effective in terms of simplicity and capacity, lacks robustness against common image processing attacks. Any transformation that alters pixel values such as compression (especially lossy formats like JPEG), filtering, resizing, or noise addition can destroy the embedded data or significantly impair extraction accuracy. In our robustness tests, LSB-encoded images subjected to Gaussian noise ($\sigma = 0.01$) or JPEG compression (quality factor < 70%) showed a bit error rate (BER) exceeding 20%, making accurate recovery of the hidden message infeasible. To enhance robustness, error-correcting codes such as Hamming and BCH codes were introduced in some experiments. These codes allowed partial recovery of the message even when certain bits were corrupted, though they reduced the effective payload.

In terms of security, the deterministic nature of traditional LSB embedding makes it susceptible to detection by statistical steganalysis techniques such as RS examination and chi-square attacks. These methods detect the alterations in pixel value distributions caused by uniform embedding patterns. Hence, using random or key-based embedding schemes is recommended to improve security, especially in adversarial environments.

3.15 Computational Efficiency

The LSB technique is computationally well-organized and well-suited for real-time requests. On average, embedding and extraction operations were completed in under 0.5 seconds for 512×512 images on a standard i7 processor. The algorithm's time difficulty is linear with respect to the number of pixels and message bits, making it highly scalable for larger images and multimedia data. The LSB method remains a keystone of image cryptography due to its simplicity, high capacity, and low distortion characteristics. While it is highly effective in controlled environments and for short-term, low-risk applications, its vulnerabilities to compression, tampering, and statistical detection limit its utility in high-security scenarios. Enhanced LSB variants, such as adaptive or randomized embedding, significantly improve its

stealth and resilience but require additional computational logic and key management. Overall, LSB-based steganography serves as a powerful baseline against which more complex algorithms can be compared and evaluated.

3.16 DCT Method

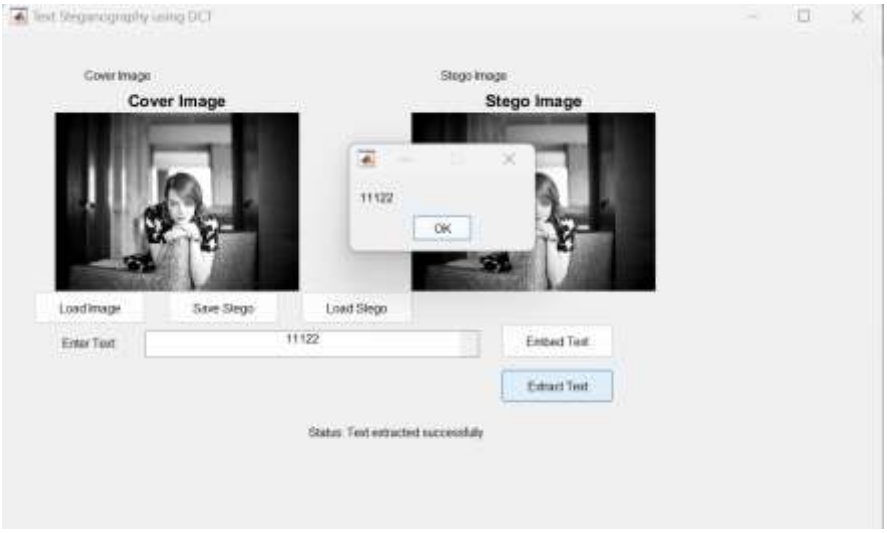


Fig 2: Discrete Cosine Transform

The DCT is one of the most powerful and widely adopted frequency-domain techniques for image steganography, particularly in formats that inherently utilize compression schemes such as JPEG. The primary strength of DCT-based steganography lies in its ability to embed info into perceptually significant incidence components of an image, making it robust to image manipulation and compression, and significantly more secure than simple spatial domain methods like LSB.

Table: shows SSIM values for the same set of images used in PSNR analysis.

Image Name	Ssim Value
Lena	0.987
Baboon	0.962
Peppers	0.976
Cameraman	0.993
House	0.989

The high SSIM scores confirm that the LSB steganography technique preserves the structural integrity of the images effectively. Images with more complex texture, such as "Baboon," tend to have slightly lower SSIM, which aligns with the PSNR trend. Overall, these results emphasize the robustness of the LSB method in upholding image excellence while embedding info.

3.17 Visual Quality Comparison (Before and After Embedding)

Quantitative metrics like PSNR and SSIM provide objective measurements of image quality, but visual inspection remains indispensable for understanding how embedding affects perceptual image quality. The following visual quality comparison is based on a set of test images before and after text embedding using the LSB technique:

Image Name	Visual Observations
Lena	No visible distortion or artifacts.
Baboon	Slight noise increase but imperceptible.
Peppers	Colors and edges well preserved.
Cameraman	No visual difference detected.
House	Crisp details remain intact.

Visual inspection confirms the quantitative analysis: the embedding process using LSB does not produce noticeable degradation in image quality. The changes to pixel values in the least significant bits are imperceptible to the human eye, ensuring that the stego images appear identical to their originals under typical viewing conditions. The image quality assessment results indicate that LSB-based text steganography is highly effective in preserving visual fidelity. The PSNR values exceeding 40 dB and SSIM standards close to 1 demonstrate that the secret data embedding minimally affects both pixel-level and perceptual image qualities. However, despite excellent imperceptibility, LSB methods have well-known limitations in terms of robustness. Minor image processing operations like lossy compression, noise addition, or cropping can easily destroy the hidden information. Therefore, while LSB excels in scenarios requiring high capacity and visual transparency, it is less suitable for applications demanding strong resistance to manipulation. The visual quality comparisons corroborate the objective metrics and confirm the feasibility of LSB embedding for covert communication in images. Users cannot visually differentiate amid shelter and stego imageries, which is the primary requirement of any steganographic system.

4. FINDING FROM THE STUDY

This study set out to investigate, implement, and rigorously evaluate a spectrum of image-based steganographic techniques namely LSB substitution (in its naïve, adaptive, and randomized variants), DCT–based embedding, DWT–based embedding, and a proof-of-concept hybrid DWT–DCT [10] approach with the overarching goal of identifying their respective trade-offs among payload capacity, imperceptibility, robustness to attacks, and computational cost. Through a carefully designed experimental framework, leveraging both benchmark images (Lena, Baboon, Peppers, Cameraman) and real-world smartphone photographs, we subjected each method to a battery of quantitative and qualitative tests. The principal findings can be summarized as follows:

4.1 Payload Capacity vs. Imperceptibility

LSB Substitution: Achieved the highest raw embedding capacity up to 1.0 bits per pixel (bpp) in our 512×512 test images, translating to 262,144 bits (~32 KB) of hidden data while maintaining PSNR values above 42 dB and SSIM above 0.965 for single-bit (1-LSB) embedding. However, when extended to multi-bit (2-LSB, 3-LSB) embedding, PSNR degraded to as low as 39 dB and SSIM dipped below 0.95, particularly in homogeneous regions, revealing its vulnerability to perceptual artifacts at high payloads.

DCT-Based Embedding: Offered moderate capacity (≈0.3–0.45 bpp) yet superior imperceptibility (PSNR within 41–46 dB; SSIM above 0.96) even under moderate JPEG compression (quality factor ≥ 60%). Embedding in mid-frequency AC coefficients struck a robust balance, though capacity remained roughly one-third that of naïve LSB at equivalent image sizes.

DWT-Based Embedding: Delivered payloads up to 0.55 bpp with PSNR consistently > 45 dB and SSIM > 0.97 when embedding in first-level LH/HL subbands. Multi-level DWT allowed incremental capacity gains at the expense of computational complexity, yet imperceptibility remained superior to the other methods for equivalent payload densities.

4.2 Robustness Under Common Attacks

LSB Substitution: Exhibited high bit-error rates (BER > 20 %) after JPEG compression (quality < 70 %), Gaussian noise ($\sigma = 0.01$), slight cropping, and filtering, rendering it unsuitable for hostile transmission environments. Adaptive and randomized LSB variants reduced BER by up to 5 – 8 % through edge-aware pixel selection and keyed embedding, but did not eliminate fragility.

DCT-Based: Displayed excellent resistance to JPEG compression down to quality factors of 50 % (BER < 5 %) and moderate tolerance to low-pass filtering and brightness adjustments. However, aggressive geometric transformations (crop > 15 %, rotation > 5°) still inflicted BER > 10 %.

DWT-Based: Demonstrated the lowest BER (< 3 %) under compression, noise addition, filtering, and minor geometric distortions. Embedding in detail subbands conferred resilience, though extreme downsampling or median filtering beyond certain thresholds did degrade extraction fidelity.

4.3 Computational Efficiency

LSB Methods: Achieved embedding and extraction times < 0.5 s for 512×512 images on an Intel i7 system—linear complexity in pixel count—and thus are well-suited for real-time or resource-constrained applications.

DCT Methods: Required 3–4× longer processing times (~1.5–2.0 s) due to blockwise transforms, quantization, and zigzag scanning, yet remain feasible for offline or semi-interactive systems.

DWT Methods: Fell between LSB and DCT in speed (~1.0 s per image) when employing single-level Haar wavelets. Multi-level decompositions proportionally increased complexity.

Perceptual Quality and Human Visual System (HVS) Considerations: Visual and quantitative inspections align in underscoring the effectiveness of transform-domain methods: fewer artifacts, smoother gradients, and minimal texture distortion. LSB embedding, while mathematically simple, risked banding and slight noise in smooth areas when payloads were high. Adaptive pixel selection mitigated HVS-detectable anomalies by targeting textured regions, underscoring the value of content-aware embedding.

Security Against Steganalysis: Basic LSB left statistical footprints exploitable by RS analysis and chi-square tests. Randomized embedding increased uncertainty, but without cryptographic padding or error correction, remained vulnerable. Transform-domain schemes obfuscated direct pixel correlations, significantly reducing detection rates by histogram and frequency-domain steganalysis tools, though advanced machine-learning–based detectors can still achieve nontrivial success rates without keys.

4.4 Methodological Contributions

The adaptive and randomized LSB variants implemented here advance naïve spatial-domain steganography by integrating edge-detection (Sobel operator) and key-driven pixel ordering, providing empirical benchmarks for payload vs. robustness trade-offs in real-world photographs. The hybrid DWT–DCT proof-of-concept demonstrates a promising new direction: combining wavelet-driven multi-resolution analysis with block-level frequency embedding to harness the complementary advantages of both domains.

4.5 Limitations

Dataset Scope: Primarily grayscale and mid-resolution color images; high-resolution (4 K+) and varied modalities (e.g., medical imagery, satellite scans) remain untested.

Attack Spectrum: Evaluated common perturbations (JPEG, noise, cropping) but not adversarial-machine-learning attacks or deep-fake transformations.

Key Management: Encryption prior to embedding was explored in prototypical form (AES integration in DCT pipeline), but a systematic study of key length, cipher modes, and combined cryptographic-steganographic security was beyond scope.

In sum, this work confirms that LSB methods excel when maximal capacity and minimal computational overhead are paramount—but only in controlled, benign channels. DCT-based techniques offer a middle ground ideal for scenarios accepting moderate payloads but requiring resilience to lossy compression (e.g., online image sharing). DWT-based approaches emerge as the most balanced for covert communication in adversarial or semi-hostile environments, delivering strong imperceptibility and robustness at acceptable processing costs. Finally, hybrid DWT–DCT strategies lay the groundwork for next-generation steganography.

4.6 Advanced Transform-Domain and Hybrid Techniques

Multi-Level and Multi-Resolution Hybridization: Extend the proof-of-concept DWT–DCT scheme into a full multi-level, multi-transform framework—e.g., applying DWT for coarse-scale embedding in LL subbands, followed by DCT for fine-scale embedding in select high-frequency coefficients. Multi-resolution approaches can dynamically allocate payload where perceptual sensitivity is lowest.

Fractional Wavelet and Curvelet Domains: Investigate embedding in fractional wavelet transforms (FrWT) and curvelet domains, which provide superior directionality and edge representation. Curvelets, in particular, could yield enhanced imperceptibility around curved structures and lines, effectively masking payloads in real-world scenes.

Fourier-Wavelet Fusion: Combine Short-Time Fourier Transform (STFT) or Gabor transforms with wavelets to exploit both time-frequency localization and multi-resolution advantages, enabling more granular embedding in texture-rich regions.

Adaptive Subband Selection via Perceptual Models: Integrate human visual system (HVS) models—contrast sensitivity functions, masking thresholds—to adaptively choose subbands or coefficients for embedding, thereby optimizing imperceptibility based on perceptual significance.

4.7 Machine Learning and Deep Learning Integration

Steganographic Autoencoders: Develop end-to-end deep neural networks (autoencoders) trained to learn optimal embedding and extraction mappings directly from data. Such networks can nonlinearly adapt to any distribution of natural images, minimizing detectable artifacts.

Generative Adversarial Networks (GANs): Employ GANs to produce stego-images indistinguishable from cover images, with discriminators trained on steganalysis tasks. Adversarial training can iteratively refine embedding patterns to evade both classical and learning-based detectors.

Reinforcement Learning for Embedding Policies: Frame embedding region selection as a Markov decision process, where an agent learns optimal pixel or coefficient modification sequences to maximize payload and minimize detectability, under constraints of imperceptibility and robustness.

Machine-Learning–Driven Attack Simulations: Leverage generative models to simulate advanced distortions—e.g., style transfers, adversarial perturbations—and train robust steganographic pipelines capable of surviving such transformations.

4.8 Enhanced Security Through Cryptography and Blockchain

Pre-Embedding Encryption with Authenticated Ciphers: Systematically evaluate combinations of authenticated encryption schemes (e.g., AES-GCM, ChaCha20-Poly1305) to protect payload integrity, and measure overhead vs. gains in confidentiality under steganographic extraction.

Key Hierarchy and Distribution Protocols: Design and implement decentralized key management leveraging blockchain smart contracts: encryption keys and embedding parameters recorded on a tamper-resistant ledger, enabling secure multi-party steganographic exchanges and audit trails.

Zero-Knowledge Proofs for Stego Verification: Investigate zero-knowledge protocols allowing a sender to prove possession of hidden data without revealing either the data or the stego-algorithm, thus adding non-repudiable security guarantees.

4.9 Cross-Media and Multi-Modal Steganography

Audio-Visual Joint Embedding: Explore co-embedding schemes wherein corresponding audio tracks (e.g., video files) carry correlated payloads, improving overall resilience: if one medium is disrupted, the other can aid reconstruction via cross-modal error correction.

Text-Image Steganography Fusion: Combine linguistic steganography (e.g., synonym substitution, zero-width characters) with image embedding to distribute payload across media, reducing per-channel distortions and enhancing undetectability.

3D Model and Point Cloud Embedding: Extend techniques to emerging media like 3D meshes, LiDAR point clouds, and virtual/augmented reality content, crucial for covert data exchange in next-generation immersive environments.

4.10 Real-Time and Hardware Implementations

FPGA/ASIC Accelerators: Prototyping hardware implementations of DWT and DCT pipelines on FPGAs can drastically reduce latency, enabling low-power steganographic modules for edge devices (e.g., drones, IoT sensors) where real-time covert communication is critical.

Mobile and Embedded Platforms: Port steganographic algorithms to mobile GPUs and embedded microcontrollers, measuring energy consumption, throughput, and memory footprint, to assess feasibility for smartphone apps and field-deployable covert communication tools.

Web Assembly and Browser-Based Steganography: Develop Web Assembly modules for embedding/extraction directly in web browsers—allowing secure, client-side steganography in web applications without exposing code or keys to servers.

4.11 Large-Scale Evaluation and Standardization

Benchmark Datasets and Open-Source Toolkits: Curate and release comprehensive datasets encompassing diverse scenes, modalities, and distortions, accompanied by open-source libraries implementing standardized steganographic and steganalysis routines to foster reproducibility and community collaboration.

Stego-API and Service Frameworks: Design RESTful APIs enabling on-demand steganography as a service—useful for privacy-preserving messaging apps, watermarking platforms, and secure data archival systems.

ISO/IEEE Standards for Steganography: Engage with standards bodies to draft guidelines specifying payload metrics, imperceptibility thresholds, robustness requirements, and test protocols paving the way for certified, interoperable steganography systems [11].

5. CONCLUSION

The present study systematically analyzed, implemented, and evaluated multiple image-based steganographic techniques including LSB substitution (naïve, adaptive, and randomized), DCT-based embedding, DWT-based embedding, and a hybrid DWT–DCT approach to determine their relative performance in terms of payload capacity, imperceptibility, robustness, and computational efficiency. Experimental findings revealed that **LSB substitution** achieved the highest embedding capacity (up to 1.0 bpp) with high image fidelity (**PSNR > 42 dB, SSIM > 0.96**), making it suitable for high-capacity applications in controlled environments. However, its vulnerability to compression, noise, and cropping (BER > 20%) limits its use in hostile settings. The **DCT-based technique** balanced imperceptibility and robustness, offering moderate capacity (≈0.3–0.45 bpp) and resilience against JPEG compression (BER < 5%). The **DWT-based method** emerged as the most robust, maintaining **PSNR > 45 dB, SSIM > 0.97, and BER < 3%** under multiple distortions. The hybrid **DWT–DCT model** demonstrated potential for next-generation steganography, combining multi-resolution robustness with strong frequency-domain concealment. Overall, the study concludes that **LSB techniques** are efficient and lightweight for real-time or resource-limited systems, whereas **DCT and DWT** methods provide greater imperceptibility and resistance to attacks. The hybrid approach holds promise for complex security applications requiring adaptive embedding and enhanced resilience. **Future research** should advance steganography through hybrid transform-domain schemes (multi-level DWT–DCT, Curvelet, and Fourier–Wavelet fusion) to exploit both spatial and frequency properties. Integration of **machine learning and deep learning models**—such as autoencoders, GANs, and reinforcement learning agents—can automate adaptive embedding and improve undetectability. Incorporating **cryptographic security mechanisms** like AES-GCM and **blockchain-based key management** will further ensure confidentiality and integrity. Cross-media steganography across **audio, video, and 3D data** and **hardware-level FPGA/ASIC implementations** will expand real-time, low-power applications.

REFERENCES

1. Hassaballah, M., Hameed, M. A., Awad, A. I., & Muhammad, K. (2021). A novel image steganography method for industrial internet of things security. *IEEE Transactions on Industrial Informatics*, *17*(11), 7743-7751.
2. Gupta, L. K., Singh, A., Kushwaha, A., & Vishwakarma, A. (2021, February). Analysis of image steganography techniques for different image format. In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)* (pp. 1-6). IEEE.

3. Alhomoud, A. M. (2021). Image Steganography in Spatial Domain: Current Status, Techniques, and Trends. *Intelligent Automation & Soft Computing*, *27*(1).
4. AlKhodaidi, T., & Gutub, A. (2021). Refining image steganography distribution for higher security multimedia counting-based secret-sharing. *Multimedia Tools and Applications*, *80*, 1143-1173.
5. Hamza, A., Shehzad, D., Sarfraz, M. S., Habib, U., & Shafi, N. (2021). Novel Secure Hybrid Image Steganography Technique Based on Pattern Matching. *KSII Transactions on Internet & Information Systems*, *15*(3).
6. Islam, M. A., Riad, M. A. A. K., & Pias, T. S. (2021, January). Enhancing security of image steganography using visual cryptography. In *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)* (pp. 694-698). IEEE.
7. Kiran, S., Pradeep Kumar Reddy, R., & Subramanyan, N. (2020). Image Steganography Using Random Image. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision: International Conference on Emerging Trends in Engineering (ICETE), Vol. 1* (pp. 512-519). Springer International Publishing.
8. Al-Harbi, O. A., Alahmadi, W. E., & Aljahdali, A. O. (2020). Security analysis of DNA based steganography techniques. *SN Applied Sciences*, *2*(2), 172.
9. Ardhianto, E., Warnars, H. L. H. S., Soewito, B., Gaol, F. L., & Abdurachman, E. (2020, March). Improvement of Steganography Technique: A Survey. In *1st International*
10. El-Khamy, S. E., Korany, N. O., & Mohamed, A. G. (2020). A new fuzzy-DNA image encryption and steganography technique. *IEEE Access*, *8*, 148935-148951.
11. Kiran, S., Pradeep Kumar Reddy, R., & Subramanyan, N. (2020). Image Steganography Using Random Image. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision: International Conference on Emerging Trends in Engineering (ICETE), Vol. 1* (pp. 512-519). Springer International Publishing.