

An Image Steganography Technique for Invisible Communication

Master in Computer Applications

by

SUHIN DUBEY

Roll: 90/MCA No: 230023

Under the guidance of

Prof. Jyotsna Kumar Mandal

Department of Computer Science and Engineering

University of Kalyani

Statutory Declarations

Name of the Candidate: Suhin Dubey

Title of the Project: An Image Steganography Technique for Invisible Communication

Degree: Masters of Computer Application (M.C.A)

Name of the Guide: Prof. Jyotsna Kumar Mandal

Registration Number: 2080039 of 2023-24

Roll: 90/MCA No. 230023

Place of Project: Department of Computer Science and Engineering,
University of Kalyani,
Kalyani, Nadia, West Bengal, India.

Declaration by the Student

I hereby declare that the work reported in the M.C.A project entitled “An Image Steganography Technique for Invisible Communication” is an authentic record of my work carried out under the supervision of Prof. Jyotsna Kumar Mandal. I have not submitted this work elsewhere for an other degree or diploma.

Place: Kalyani

Date: ____/____/____

Signature of Student

ACKNOWLEDGEMENT

I would like to express my deep sense of gratitude and indebtedness to the many individuals who contributed to the success of my project. Foremost, I am fortunate to have the guidance and support of my supervisor, Prof. Jyotsna Kumar Mandal. His uncountable advice and encouragement played a pivotal role in the systematic completion of my project, and I am confident that his influence will resonate throughout my career.

I extend my thanks to the Head of the Department, Prof. Priya Ranjan Sinha Mahapatra, as well as Prof. Anirban Mukhopadhyay, Prof. Utpal Biswas, Prof. Kalyani Mali, Dr. Debabrata Sarddar, Mr. Sukanta Majumdar, Mr. Jaydeep Paul and Mrs. Shrabanti Kundu from the Department of Computer Science and Engineering at the University of Kalyani. Their support and cooperation were invaluable.

I am grateful to the entire Department of Computer Science and Engineering at the University of Kalyani for their assistance and collaborative efforts.

No success would be complete without acknowledging the role of my parents. Their unwavering support and encouragement have been the driving force behind my education and achievements. They made it their life's mission to ensure I stayed focused on my goals. With their blessings and wise words, I found the strength to overcome challenges. I deeply appreciate their foresight and simplicity, which have been instrumental in helping me achieve my aspirations.

Suhin Dubey

Date: 25.01.25

Contents

1. Introduction	
1.1 Basics of Image Processing.....	01
1.2 What is Steganography?.....	02
1.3 What is Image Steganography?.....	03
1.4 Why it is important?.....	05
2.	
2.1 Problem Formulation.....	07
2.2 Project Goals.....	08
2.3 Evaluation Criteria.....	08
2.4 Performance Analysis.....	09
3. Methodology.....	09
3.1 Library Used.....	11
3.2 Encoding Algorithm.....	11
3.3 Decoding Algorithm.....	13
3.4 Image Quality Metrics Calculation.....	14
4. Result.....	16
5. Conclusive Discussion & Future Scope.....	18
5.1 Conclusion.....	18
5.2 Future of Work.....	19

List of Figures:

1. Results table.....	16-17
-----------------------	-------

CHAPTER 1

1. Introduction

Image steganography, the practice of concealing information within a cover image, is the focus of this project. This tool enables the encoding of secret messages into image files while preserving high visual quality. It also provides functionality for decoding hidden messages from encoded images and calculating key image quality metrics. The tool offers a user-friendly interface for encoding messages into images, decoding hidden messages, and calculating image quality metrics, including Mean Squared Error (MSE), Image Fidelity (IF), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM) for comparing original and encoded images. By carefully selecting the bits to modify within the image, the tool minimizes disruption to the original pixel values, thus prioritizing the visual integrity of the stego image. This project serves as a demonstration of image steganography's potential for secure data hiding, while acknowledging the ethical considerations and the potential for misuse of such techniques.

1.1 Basic of Image Processing

Image processing is a field of study that involves analyzing, manipulating and enhancing digital images using various techniques and algorithms. It is a fundamental component of computer vision and has applications in numerous fields, including medicine, surveillance, remote sensing and entertainment.

Digital images are represented as grids of picture elements, or pixels, each containing numerical values corresponding to color or intensity (Image Representation). Images are acquired through devices like cameras, scanners, and sensors, often undergoing preprocessing for noise

reduction or calibration (Image Acquisition). Image Enhancement techniques, such as adjusting brightness, contrast, color, noise reduction, and edge sharpening, improve visual quality. Image Restoration aims to recover degraded images through noise removal, de-blurring, and inpainting. Image Compression reduces file size, using lossless methods to preserve all data or lossy methods to achieve smaller sizes with some quality loss. Image Segmentation partitions images into meaningful regions based on characteristics like color, texture, or intensity, crucial for object recognition. Image Feature Extraction identifies distinctive attributes like edges, corners, textures, and color histograms for analysis and recognition. Image Filtering modifies images via mathematical operations on pixels or pixel neighborhoods, including blurring, sharpening, and noise reduction. Image Transformation techniques alter spatial properties through scaling, rotation, and flipping, used for alignment and geometric correction. Finally, Image Analysis extracts information through tasks like object detection, recognition, tracking, and measurement. These fundamentals form the basis of a broad field encompassing advanced topics like image registration, morphological operations, and machine/deep learning-based image analysis.

1.2 What is Steganography?

Steganography is the practice of concealing information within a cover medium (like images, audio, or video) to hide its very existence. Unlike cryptography, which scrambles data, steganography aims for undetectability. The process involves embedding a secret message into the cover medium, creating a stego medium that is perceptually indistinguishable from the original.

Key components:

- **Cover Medium:** The carrier (e.g., image).
- **Secret Message:** The hidden information.
- **Stego Medium:** The cover with the hidden message.

Common techniques:

- **Least Significant Bit (LSB) Insertion:** Modifying the least significant bits of the cover's data.
- **Spatial/Frequency Domain Methods:** Manipulating pixel values or frequency components.
- **Transform Domain:** Uses transforms like wavelet transform.
- **Model-Based:** Employs machine/deep learning.

Effective steganography prioritizes:

- **Imperceptibility:** Undetectable to human senses.
- **Capacity:** Amount of data that can be hidden.
- **Robustness:** Resistance to modifications (e.g., compression).
- **Security:** Difficulty for unauthorized extraction.

Applications include secure communication, digital watermarking, and data hiding. Steganography can be combined with cryptography for stronger security: encrypting the message before hiding it.

1.3 What is image steganography?

Image steganography is the art and science of concealing information within a digital image in such a way that its presence is undetectable to the human eye. It leverages the inherent redundancy in digital image data to embed secret messages without causing noticeable visual distortions. Unlike cryptography, which scrambles data to make it unreadable, steganography hides the very existence of the message.

The process involves embedding a secret message (which can be text, another image, audio, or other digital data) into a cover image. The resulting image, known as the stego-image, appears visually identical to the original cover image. The recipient, knowing the steganographic method used, can then extract the hidden message.

Key aspects of image steganography include:

- **Cover Image:** The original image used to hide the message.
- **Secret Message:** The information to be concealed.
- **Stego-Image:** The resulting image containing the hidden message.
- **Steganographic Algorithm:** The method used to embed and extract the message.

Common techniques include:

- **Least Significant Bit (LSB) Insertion:** The most basic method, where the least significant bits of pixel values are replaced with the message bits.
- **Spatial Domain Methods:** Directly manipulate pixel values.
- **Frequency Domain Methods:** Embed data in the frequency domain of the image (e.g., using Discrete Cosine Transform).
- **Transform Domain Techniques:** Similar to frequency domain but using different transforms.

Effective image steganography aims for:

- **Imperceptibility:** The stego-image should be visually indistinguishable from the cover image.
- **Capacity:** The amount of data that can be hidden without causing noticeable distortions.
- **Robustness:** Resistance to attacks and image processing operations (e.g., compression, cropping).
- **Security:** Difficulty for unauthorized detection and extraction.

Image steganography has applications in secure communication, digital watermarking, and data hiding. It can also be combined with cryptography for enhanced security by encrypting the message before embedding it.

1.4 Why is it important ?

Image steganography is important for several reasons, primarily related to security, privacy, and data integrity:

- **Confidential Communication:** It allows individuals or organizations to communicate secretly without raising suspicion. Unlike encryption, which makes data unreadable, steganography hides the very existence of the communication. This can be crucial in situations where encryption might attract unwanted attention, such as in oppressive regimes or during covert operations.
- **Data Protection and Integrity:** Steganography can be used to embed watermarks or digital signatures within images to protect copyright and verify authenticity. This can help prevent unauthorized copying or modification of digital content.
- **Circumventing Censorship:** In regions with strict censorship, steganography can be used to bypass restrictions and distribute information freely.

Hidden messages within seemingly innocuous images can be a way to share news, opinions, or other content that would otherwise be blocked.

- **Secure Storage and Transmission of Sensitive Data:** Steganography can be used to hide sensitive data, such as medical records, financial information, or personal identification, within images. This adds an extra layer of security, as even if the image is intercepted, the hidden data remains protected.
- **Covert Communication in Espionage and Intelligence:** Steganography has long been used in espionage and intelligence operations to transmit secret messages without detection. This is

because it is much harder to prove that steganography has been used than it is to prove that encryption has been used.

- **Bypassing Firewalls and Intrusion Detection Systems:** Some steganographic techniques can be used to bypass firewalls and intrusion detection systems by embedding malicious code or data within images. This is a more malicious use of steganography, but it highlights its potential for bypassing security measures.

It's important to note that while steganography has legitimate uses, it can also be misused for malicious purposes, such as hiding malware or distributing illegal content. Therefore, it's essential to be aware of both the benefits and risks associated with this technology.

CHAPTER 2

2.1 Problem Formulation

This project aims to develop a user-friendly image steganography tool with the following core functionalities:

- **Data Hiding:** Implement an algorithm to encode a secret message into a digital image (cover image) in a way that conceals its presence.
- **Data Extraction:** Develop a corresponding algorithm to accurately extract the hidden message from the stego-image (the encoded image) for authorized retrieval.
- **Image Quality Assessment:** Integrate methods to evaluate the perceptual quality of the stego-image compared to the original cover image. This will involve calculating and displaying relevant metrics such as Mean Squared Error (MSE), Image Fidelity (IF), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).
- **User Interface:** Design and implement a simple and intuitive menu-driven interface to facilitate user interaction with the encoding, decoding, and metric calculation functionalities.

2.2 Project Goals:

- To provide a straightforward and accessible tool for basic image steganography.
- To demonstrate the principles of data hiding within digital images.
- To quantify the impact of the steganographic process on image quality using established metrics.

2.3 Evaluation Criteria:

- **Imperceptibility:** The visual similarity between the cover image and the stego-image will be assessed using PSNR and SSIM. Higher values indicate better imperceptibility.
- **Capacity:** The amount of data that can be successfully hidden within the image will be measured.
- **Accuracy:** The accuracy of the message extraction process will be evaluated by comparing the extracted message with the original hidden message.
- **Usability:** The ease of use and clarity of the user interface will be assessed through user observation or feedback.

This project focuses on the implementation and evaluation of a basic image steganography tool and its impact on image quality.

2.4 Performance analysis:

PSNR

PSNR is for estimating the imperceptibility of the reconstruction of an image. The PSNB higher PSNR value indicates a higher signal to noise ratio.

The formula to calculate PSNR is as follows:

$$\text{PSNR}_{\text{db}} = 10\log_{10}(255^2/\sqrt{\text{MSE}})$$

Here,

MAX=255, represents the maximum possible pixel value in the image
Example: 255 for 8-bit grayscale or 255, 255, 255 for 24-bit color images.

MSE stands for Mean Squared Error, which is the average squared difference between the pixel values of the original and reconstructed images. MSE is a parameter used to find the signal loss. The nature of the obtained image is better if MSE is lesser.

MSE can be calculated by,

$$\text{MSE} = 1/M*N \sum \sum [f(p,q) - g(p,q)]^2$$

PSNR, or Peak Signal-to-Noise Ratio, is a metric commonly used in image and video processing to evaluate the quality or fidelity of a reconstructed or compressed image compared to the original. It measures the ratio between the maximum possible power of a signal (in this case,

the original image) and the power of the noise or distortion present in the reconstructed image. The higher the PSNR value, the lower the distortion or noise in the reconstructed image compared to the original. A higher PSNR value generally indicates better image quality.

SSIM (Structural Similarity Index Measure)

- **Purpose:** Measures the perceptual similarity between two images by considering luminance, contrast, and structure.
- **Key Idea:** SSIM aims to capture how humans perceive image quality, focusing on structural information rather than just pixel-wise differences.
- **Range:** -1 to 1, where 1 indicates perfect similarity.
- **Advantages:** More closely aligns with human visual perception compared to metrics like MSE.

IF (Image Fidelity)

- **Purpose:** A metric that assesses the fidelity of an image, often used in image processing and compression.
- **Calculation:**
 - Derived from Mean Squared Error (MSE).
 - Generally, $IF = 1 - (MSE / \text{mean}(\text{original_image}^2))$.
- **Interpretation:**
 - A value closer to 1 indicates higher fidelity, meaning the processed image is more similar to the original.
 - Lower values indicate greater distortion or loss of information.

CHAPTER 3

3 Methodology: LSB Image Steganography with Hash-Based Bit Selection

This project implements image steganography using Least Significant Bit (LSB) substitution in the spatial domain. The core components are:

3.1 Libraries Used

- **PIL (Pillow):** For image loading, manipulation, and saving.
- **NumPy:** For numerical operations, especially for calculating image quality metrics.
- **scikit-image (skimage):** Specifically, the `structural_similarity` function for calculating SSIM.
- **OpenCV (cv2):** For image reading (using `cv2.imread`) and color space conversion (for SSIM calculation).

3.2 Encoding Algorithm (`encode_message` function)

3.2.1 Input:

- `image_path`: Path to the cover image.
- `message`: The secret message to be hidden (string).

- output_path: Path to save the stego-image.
- step: A parameter used in the hash function (currently fixed at 1).

3.2.2 Steps:

- Image Loading and Preprocessing:** The cover image is opened using `PIL.Image.open()` and converted to RGB mode if necessary. A copy of the image is created to avoid modifying the original.
- Message Conversion:** The message string is converted to its binary representation. Each character is converted to its ASCII code, which is then represented as an 8-bit binary string. A delimiter ("00000000") is appended to the binary message to mark its end.
- Bit Position Selection (Hash Function):** A simple hash function, $\text{get_bit_position}(x, y, \text{step}) = (x + y) \% \text{step}$, is used. This function takes the pixel coordinates (x, y) and the step value as input. It determines which bit within each color channel of the pixel will be used for embedding. With $\text{step} = 1$, it always uses the least significant bit. step can be changed according to you.
- LSB Substitution:** The algorithm iterates through each pixel (x, y) and each color channel (R, G, B). For each channel:
 - The bit position is determined using the hash function.

- ii. The LSB at the chosen position is cleared using a bitwise AND operation ($\text{pixel}[n] \& \sim(1 \ll \text{bit_pos})$).
 - iii. The corresponding bit from the binary message is then inserted into the cleared LSB using a bitwise OR operation ($\text{pixel}[n] | (\text{int}(\text{binary_message}[\text{data_index}]) \ll \text{bit_pos})$).
- e. Stego-Image Saving:** The modified image is saved to the specified `output_path`.
- f. Metrics Calculation:** The `calculate_metrics` function is called to evaluate the stego-image quality.

3.3 Decoding Algorithm (`decode_message` function)

3.3.1 Input:

- `image_path`: Path to the stego-image.
- `step`: The same step value used during encoding (must be known for correct decoding).

3.3.2 Steps:

- i. **Stego-Image Loading:** The stego-image is opened using `PIL.Image.open()`.
- ii. **Bit Extraction (using the same hash function):** The algorithm iterates through the pixels using the same hash function used during encoding. The bit at the position determined by the hash function is extracted from each color channel using a bitwise AND operation after right shifting ($(\text{pixel}[n] \gg \text{bit_pos}) \& 1$).

- iii. **Delimiter Check:** The extracted bits are appended to a binary string. The algorithm continuously checks for the delimiter ("00000000"). When the delimiter is found, the decoding process stops.
- iv. **Binary to Text Conversion:** The binary string (excluding the delimiter) is converted back to a text string by grouping the bits into 8-bit chunks and converting each chunk to its corresponding ASCII character.
- v. **Message Return:** The decoded message is returned.

3.4 Image Quality Metrics Calculation (calculate_metrics function)

3.4.1 Input:

- original_img_path: Path to the original image.
- encoded_img_path: Path to the stego-image.

3.4.2 Steps:

- i. **Image Loading and Type Conversion:** Images are read using `cv2.imread()` and converted to floating-point representation (`astype(float)`) for accurate calculations.
- ii. **Mean Squared Error (MSE):** Calculated as the mean of the squared difference between the pixel values of the original and stego-images.
- iii. **Image Fidelity (IF):** Calculated as $1 - (\text{MSE} / \text{mean}(\text{original}^2))$.

- iv. **Peak Signal-to-Noise Ratio (PSNR):** Calculated using the formula: $20 * \log_{10}(\text{MAX_PIXEL}) - 10 * \log_{10}(\text{MSE})$, where MAX_PIXEL is 255.
- v. **Structural Similarity Index Measure (SSIM):** Calculated using `skimage.metrics.structural_similarity` after converting both images to grayscale using `cv2.cvtColor()`. The `multichannel=False` argument is used since the images are grayscale.





Key Aspects

- **LSB Substitution:** This is a simple spatial domain technique that offers high capacity but is vulnerable to attacks.
- **Hash Function:** The `get_bit_position` function provides a simple form of bit selection within the pixels. More sophisticated methods could be used for improved security.
- **Delimiter:** The use of a delimiter is crucial for correctly determining the end of the hidden message during decoding.
- **Metrics:** The included metrics provide a quantitative evaluation of the steganographic process's impact on image quality.

CHAPTER 4

4. Results

In the below table all the real images and stego images are shown with the MSE, IF, PSNR, SSIM values. For all the picture the hidden message is same which is a 100 word message “Lorem ipsum, dolor sit amet consectetur adipisicing elit. Reprehenderit blanditiis est ipsum adipisci vero fugiat modi facere labore! Consequuntur ex, accusamus quis ut unde doloremque qui quod ipsa ducimus officiis commodi eius illum quae, adipisci hic? Aspernatur repellat ipsam obcaecati officiis, quidem fugit voluptate autem error nisi aperiam alias mollitia officia veniam maxime qui inventore, quam minima id quod dicta libero itaque, eveniet nesciunt quia. Voluptates distinctio itaque commodi asperiores repellat earum debitis doloribus provident esse! Illum et aliquid odit quam error! Alias harum sunt aspernatur necessitatibus minus eius corrupti vel, hic dolorem vitae accusamus a reiciendis eveniet expedita assumenda.”

Cover Image	Stego Image	MSE	IF	PSNR	SSIM
		0.0102	1.0000	68.06 dB	1.0000
		0.0040	1.0000	72.06 dB	1.0000







		0.0242	1.0000	64.30 dB	0.9999
		0.0012	1.0000	77.47 dB	1.0000
		0.0061	1.0000	70.25 dB	1.0000

Fig1: Results for the given image

CHAPTER 5

5. Conclusive Discussion and Future Scope

5.1 Conclusion

This project presented the implementation of an image steganography tool using the Least Significant Bit (LSB) substitution technique in the spatial domain. A simple hash function was employed to select the embedding bit within each pixel, adding a slight layer of complexity to the basic LSB method. The tool provides a user-friendly interface for encoding text messages into cover images and subsequently decoding them. Furthermore, the project incorporated essential image quality metrics, including MSE, Image Fidelity (IF), PSNR, and SSIM, to quantify the impact of the embedding process on the stego-image. The calculated metrics generally indicated a high degree of similarity between the cover and stego-images, demonstrating the effectiveness of the LSB method in maintaining imperceptibility, especially for small message sizes. This work serves as a practical demonstration of image steganography principles and provides a basis for exploring more robust and secure techniques, such as frequency domain embedding or deep learning-based methods.

5.2 Future Scope

This project developed a functional image steganography tool based on LSB substitution with a hash-based bit selection scheme. The

tool successfully encodes and decodes text messages within images while providing a quantitative assessment of the resulting stego-image quality through MSE, IF, PSNR, and SSIM. The results confirm that LSB embedding can achieve reasonable imperceptibility for small payloads. However, this method is known to be vulnerable to certain steganalysis attacks. Future work could focus on enhancing the security and robustness of the steganographic scheme by exploring:

- **Adaptive LSB Embedding:** Embedding data based on image complexity.
- **Frequency Domain Techniques:** Using DCT or wavelet transforms for more robust embedding.
- **Deep Learning-Based Steganography:** Employing CNNs or GANs for improved imperceptibility and capacity.
- **Improved Hash Functions or Pseudo-Random Number Generators:** For more secure bit selection.
- **Resistance to Steganalysis Attacks:** Evaluating the system against various steganalysis methods.

This project provides a valuable educational tool and a stepping stone for further research into more advanced and robust image steganography techniques.