Cyber Security and Applications 3 (2025) 100069

Contents lists available at ScienceDirect

Cyber Security and Applications

journal homepage: http://www.keaipublishing.com/en/journals/cyber-security-and-applications/

An effective steganographic technique for hiding the image data using the LSB technique

Rasmita Panigrahi, Neelamadhab Padhy*

School of Engineering and Technology, Dept. of Computer Science and Engineering, GIET University, Gunupur, Gobriguda, Odisha, India

ARTICLE INFO

Keywords:
Steganographic techniques
Encode, Decode
Least significant bits
Stego image watermarking
QR Code
Embedding

ABSTRACT

Steganography is the art and science of writing secret messages so that neither the sender nor the intended recipient knows there is a hidden message. Data hiding is the art of hiding data for various reasons, such as keeping private data, secure, confidential data, etc. With increasing data exchange over a computer network, information security has become a significant issue. There are many methods used for data hiding, and steganography is a well-known technique. Steganography is the art of invisible contact and science. Steganography is the process through which the presence of a message can be kept secret. The objective of this paper is to hide data using the LSB (Least Significant Bit) technique into images that can be detected only by the specified user. We have developed a user-friendly GUI such that it can be used with the utmost ease. This paper is motivated to hide the message stated by the user in the dialog box given within the picture. The secret text is converted to the ciphertext to make it more stable. The sender selects the cover image, and it is used to generate the secured Stegno image, which is identical to the cover image. With the support of a private or public communication network, on the other hand, the stegno image can be saved and sent to the designated user, i.e., the recipient downloads the stegno image and can retrieve the secret text concealed in the stegno image using that same application. As for the watermarking, we have visible and invisible we have used the same LSB technique. In visible watermarking, text or image is embedded in the cover image, which can be noticed easily. As for invisible watermarking, some specific text is inserted into an image, and while retrieving it, it generates a QR code, which can be scanned to get the watermarked text. We used the three different types of cover images i.e. Gray, and RGB also estimated the performance metrics. SNR, MSE, and PSNR the three performance metrics are used, and found that PSNR achieved good results i.e.,71.4733. The RGB image with the hidden text is achieved up to 77.6697

1. Introduction

The growth of electronics has led to the digitization of all documentation, videos, and audio. This scenario has increased the need for the security and reliability of documents, audio, or videos to maintain privacy, avoid piracy, and mass production reproduction. This varies from organization to organization or individual to individual. Cryptography and steganography are used to achieve the above requirements. Today's digital media provides reliable and easy ways to edit data, and these data are needed to be delivered safely over a secure network. Steganography is a process of hiding data in the bits of cover objects like a graphic or an audio file. Steganography provides a safe way to communicate privately because the presence of information in the cover item is challenging to detect. Media files are suitable for steganography due to their large size. Data can be inserted into a file using human perception. The audio file uses frequency masking on the tones with the same frequencies, and the listener cannot hear the masked quieter tone. The primary goal of steganography is to facilitate secret communication in an organization or between two users. An anonymous person cannot access the information by having a look at the cover file. Steganography is different from cryptography. It is employed to hide the data inside another cover object, whereas, in cryptography, data is encrypted. If the information is extracted in cryptography, then the exact data would again be difficult to obtain.

Cryptology is the way to send the encrypted message further needs to be decrypted in the form of the original. Since the attacker would not recognize the hidden message during a communication, this mechanism ensures confidentiality.

However, the unreadable type can alert adversaries to the fact that a hidden communication is taking place, making the data vulnerable to active attacks such as data blocking or alteration. As a result, steganography mechanisms have been introduced as a way to get around the lim-

Peer review under responsibility of KeAi Communications Co., Ltd.
* Corresponding author.
E-mail address: dr.neelamadhab@giet.edu (N. Padhy).

https://doi.org/10.1016/j.csa.2024.100069
Received 6 May 2024; Received in revised form 13 July 2024; Accepted 2 August 2024
Available online 14 August 2024

itations of encryption techniques. Steganography is the process of embedding hidden data into a cover medium to create a stego medium. The stego medium must be made to look exactly like the cover medium.et al., [13] used the DCT technique in steganography. They exhibited the correlation and concluded, that if the correlation is high then proved the stronger compaction. They discussed more about the data hidden concept. They also discussed how segmentation helpful for steganography

Major Contribution:

1. We have developed an LSB-based steganographic technique
2. Created one GUI-based application and Outlined the process of hiding a user-specified message within an image, converting it into cipher text for enhanced security, and generating a stegno image that appears identical to the original cover image.
3. Described properly the process of securely transmitting the stegno image over private or public communication networks to the designated recipient, who can then retrieve the hidden message using the same application.
4. Used the performance metrics, with results demonstrating high image quality retention.

This article contains 8 sections. Section 1 provides a details background of steganography and their insights. Sections 2 and 3 discuss the state-of-the-art of steganography and its types. Sections 4 and 5 give the outline of the digital watermarking system and its procedure. Section 6 discusses how the system is designed for the application and Section 7 explained briefly the algorithms required to be developed for the application. Sections 8 and 9 discussed the system design and performance metrics evaluation

2. Literature review

Alma, Wardhani, et al. [1], have developed the program used to send the original bar code and generate the stegoimage barcode using the least significant steganography technique. Such two images were compared and checked with performance metrics like MSE, RMSE, eans, and PNR. Ramesh, M. et al. [2]. has developed the hybrid steganography method, which allows reading the QR code and converting it into the discrete wavelet transformation form. In this article, they have used two different techniques; these are frequency domain encoding and decoding. Their objective was to hide the information. Chitradevi, B. et al. [3]. discussed the data hiding techniques by using LSB steganography. This paper presents a short idea for image steganography, which uses the LSB algorithm to hide the data in an image. Zhu, J. et al. [4] have used the deep learning technique for data hiding. They have discussed the adversarial, which helps to improve the encoded image video appearance. The novel Convolutional Network is used to learn to encode a rich by using invisible disturbance Relevant quantities of details. Bal, S.et al. [5]. discussed the image watermarking technique by using LSB and bit pair similarity. They have the symmetric key cryptography technique for data security purposes. Jung, K. et al. [6]. discussed the method of hiding the data using the LSB technique also discussed a semi-reversible data-hiding method. The LSB technique is used to insert secret data. Bamatraf, A. et al. [7]. suggested a new kind of algorithm called embedding and extraction. They have used peak signal-to-noise ratio (PSNR) for reading the quality images. Duan, X. et al. [8] have discussed about the PSNR and SSM. On their finding, they obtained values around 40 dB and 0.96

They used both cryptography and deep neural network techniques to enhance image steganography. Pichardo-Méndez et al. [9] studied the LSB mechanism to hide the information from the digital image. The author YiZhang et al., [10]. They have used the proposed algorithm where more statistical inference is used. They have discussed how the stego image suffers during JPEG compression. Due to this reason, they have developed a methodology called “compression-resistant Domain Constructing + RS-STC Codes, “ which resolves the JPEG compression issues. Swain, G. et al. [11] proposed an algorithm that allows handling high-capacity stegno images which is the LSB technique. Their approach is to divide the image into non-overlapped 3 × 3-pixel blocks after they have used the LSB technique for every block subsequently. Nolkha, A. [12] has done a systematic literature survey. They mainly focus on how LSB is used in the stegoimage. They have taken a color image because it holds a large amount of secret data. Speech steganography techniques are divided into two categories based on the hiding domain: time-domain schemes and frequency-domain schemes. Secret messages are encoded in time-domain speech samples in a time-domain scheme. On the other hand, frequency-domain systems embed the hidden message in a specific part of the speech signal’s frequency spectrum. Several speech concealment systems have recently been published in the literature [14-16].

3. Steganography and types

Steganography attempts to hide information in the covering data in such a way that non-participating persons can not recognize the presence of this information by analyzing the detection of data. In contrast to watermarking, steganography is not intended to prevent opponents from removing or changing the hidden message found in the cover data from the secret material. Still, it emphasizes that it is steganography is especially essential for applications where confidential information can not be encrypted to secure communication.

3.1. Text steganography

It is a steganography process where information is concealed in text files. The data is hidden by manipulating the spaces and tabs after nth letters or nth lines. Text steganography is attained by changing the text’s format or modifying the characteristics of the characters. This approach aims to create change, which is reader-friendly and unnoticeable. These characteristics are a bit conflicting with each other. To achieve steganography, various methods can be used for steganography, and these are

- Line-Shift Coding: A text file can be modified by vertically shifting the positions of the lines to encode the file uniquely.
- Word-Shift Coding: A text file can be modified by horizontally shifting the position of the words in each line to encode the file uniquely.

3.2. Image steganography

Steganography technique in which information is hidden inside an image. The image used for protecting the data is called a cover image. In this technique, the intensities of the pixels are modified according to the secret data. In digital steganography, images are preferred to hide data because it consist of many bits that digitally represent the image.

3.3. Ssteganography in image

In image steganography, the secret message or data is hidden inside the images by altering their visible properties. For example, the cover image can be modified in the noisy with color variations to remain unnoticeable of the modifications. The Least Significant Bits(LSB) method is the most common method to alter the images, and it also consists of processes like masking, filtering, and transformation. These methods are used according to image quality. Image steganography is performed for images in such a way that the embedded data can be retrieved easily. Image steganography refers to embedding data, i.e., text, images, or audio or video files. Our project aims to use image steganography to use spatial domain techniques to achieve text hiding and visible and invisible watermarking. Only a particular decoding technique will retrieve the secret message. Python incorporates both the encoding and decoding techniques.

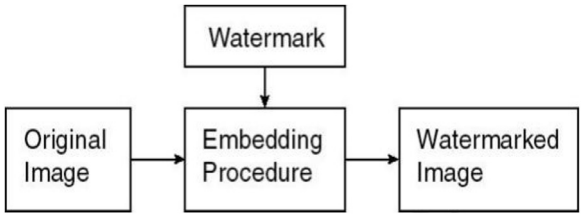


Fig. 1. Watermark encoding procedure.

4. Digital watermarking

Digital watermarking or watermarking is a way and method to hide information, and they can be text or media files. Watermarking is a process in which a message is hidden in a digital object like text, pictures or videos, or images and can be retrieved Steganography is another type of watermarking where information in the digital object is shielded existence is not visible. The currency of India is an example of watermarking. An image goes through an embedded processor with a watermark in the general watermarking process, producing a watermarked image.

5. General procedure for watermarking

Watermarking is a process in which data is called a watermark or tag or digital signature or label into a digital object so that watermark is discovered and fetched later to make an attestation about the object. The object may be a video or image or audio. Generally, the watermarking algorithm consists of three parts:

- Watermark
- Encoding algorithm
- Decoding algorithm

Watermarks are typically used for copyright protection of objects by their owners. Every owner has a unique watermark or and can use different watermarks for different objects. The encoding algorithm embeds the watermark on the object, and the decoder algorithm authenticates the object by determining the integrity and owner (Fig. 1).

5.1. Encoding

The below-mentioned Fig. 2 shows that the original image will be converted into the watermarked image with the help of the embedding procedure. The Figure below describes the watermark encoding process:

Let us denote an image by I, a signature by $S = \{s1, s2, \ldots sn\}$ labeled the watermarked image.

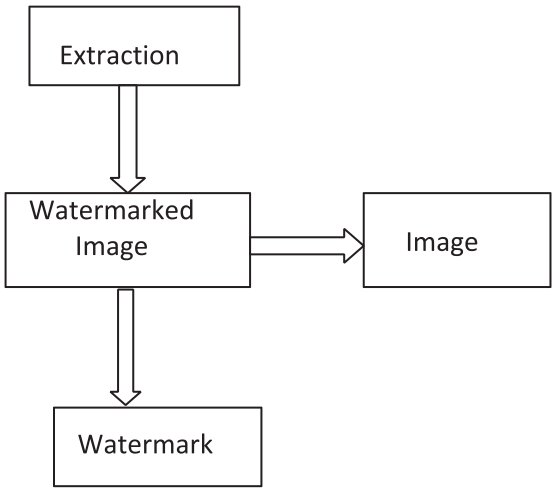


Fig. 2. Watermark decoding process.

E is an encoder feature, it takes an image I and a signature S, and it produces a new model called a watermarked image I', that is to say.

$E(I, S) = I'$.

5.2. Decoding

A decoder function D takes an image J (J may be a watermarked or unwatermarked image, and probably corrupted) whose ownership is to be determined and recovers from the image a signature S'. An additional picture can also be used in this process that also the original and unwatermarked version of J. This is because specific coding systems can use the authentic images in the watermarking process to ensure extra robustness against deliberate and unintentional pixel corruption.

Mathematically.

$D(J, I) = S'$

The below mentioned Fig. 2 describes the watermark decoding process

The Watermarking method used can involve unique approaches depending upon the way the watermark is embedded and relies on the behavior of the watermarking algorithm. In some cases, the actual watermark is extracted in the same form, called watermark extraction. In other schemes, only the presence of the specific watermarking signal is detected, called watermarking detection.

6. System design

The below-mentioned Fig. 3 represents the schematic diagram of visible watermarking techniques. A visible watermark is a transparent overlay in a picture and can be seen, by the viewer. Visible watermarking is used to display ownership and to protect copyrights. Whereas an invisible watermark is inserted in the data so that the changes made to the pixel values are not seen perceptually. Invisible watermarks are used as evidence of ownership and for detecting misappropriated

The above-mentioned Fig. 4 is the complete layout of our application. The working principle of the application is as below:

The user will supply the source file from the editor, then the python interpreter will compile it into the byte conde through the VM connecting to the different libraries and finally, it gives the output.

7. BRIEF algorithm implementation

7.1. Least significant bit (LSB)

The least significant bit is the most commonly used technique in image steganography. LSB technique involves altering pixel intensities by a small value, which results in a negligible change in the image, which remains unnoticeable by the human eye.

There are two different methods to achieve image steganography:

- Spatial method
- Transform method

But for our project, we are using the Spatial method.

7.2. Spatial method

In the LSB method, the spatial method is most commonly used. It is a common, simple process for hiding or embedding information on an object or cover file. In image steganography, the LSB method is used for each pixel. An image has three components (Red, Green, and Blue). The information of the pixel is stored in an encoded format in one byte. To hide data in an image, the first criterion is that the text size should be less than equal to the size of the image used as a cover image. The resulting image in this process is vulnerable to cropping and noise. In this method, the most significant bits of the text are stored in the least significant image bits. Pixels in an image are stored as bits in the image.

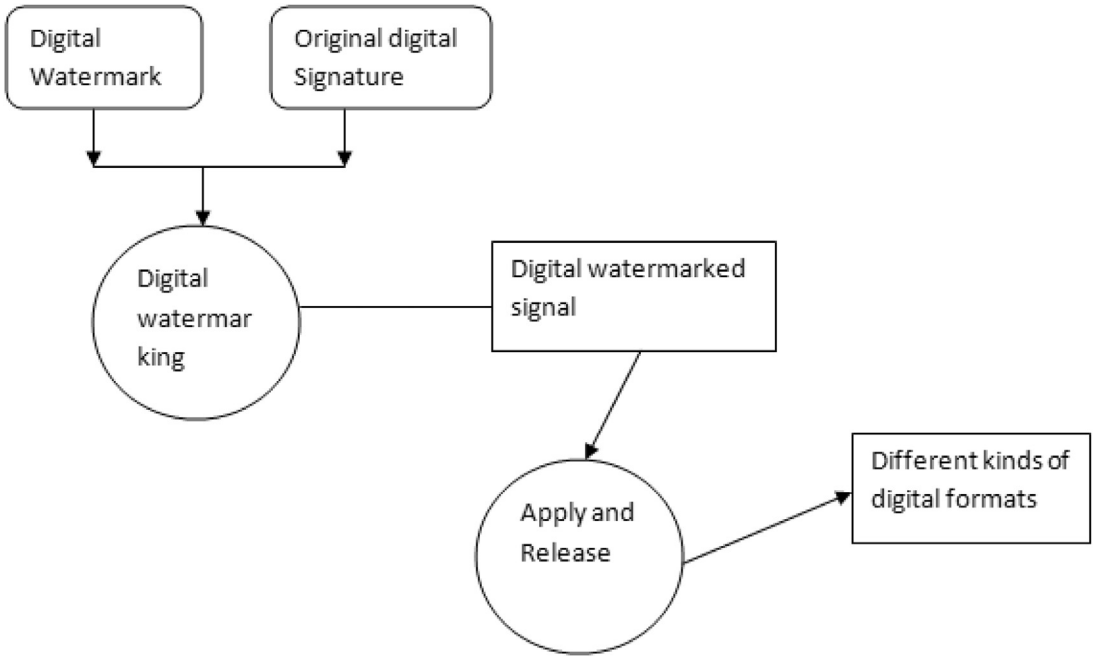


Fig. 3. Schematic Design for visible watermark.

Each pixel is 8 bits in the gray image, and for a color image, each pixel is 24 bits. Our project is only applicable to colored images. The Human Visual System cannot detect modifications of pixel intensities when the least significant bit is modified. LSB is used as an advantage to hide information in the image with minimum differences in the picture.

7.3. Encoding Procedure

Table 1
Pseudocode for encoding procedure.

1. Extracting all the pixels from the given image.
2. Extract all the characters from the given text.
3. Next phase, converting the given text into cipher text for more security using a specific key.
4. Take 3 pixels and one character from the ciphertext. Each character is of 8 bits, so replacing the LSB of each color, i.e., RGB, with every bit of the character.
5. If all characters are present, the LSB of the third color of the third pixel is 0, or 1 if all the characters are left.
6. Each character is stored in three pixels, and the LSB of the third pixel determines the end of the message.
7. As the text embedding finishes, a stego image is generated, which is identical to the cover image.

7.4. Decoding Procedure

Table 2
Pseudocode For Encoding Procedure.

1. Extracting all the pixels from the stego image.
2. Taking 3 pixels at a time and storing the LSB of each color in an array.
3. Checking third-pixel LSBs. If it is 0 means, it is the end of the text message else continue to store the LSB in the array.
4. After storing all the LSB values, it is transformed into a ciphertext character.
5. Then the ciphertext is converted back to the original text using the same previous key.

During implementation, we have used the databases(MySQL, SQLite, Firebird, Oracle) and the test frameworks (Robot, PyTest, Unittest, DocTest, Nose2, Testify), web scrapingMechanize, BeautifulSoup), Image processing (Scikit-image, OpenCV, SciPy, Pillow). For Data Analytics, we have used toolkits like -NLTK, SciPy, and NumPy, apart from the Machine learning algorithms used. The above-mentioned Tables 1 and 2 for the pseudocode for encoding and decoding procedure.

The simple techniques hide the bits of the text directly into the LSB of the image or cover image in a linear sequence. Modulation of the LSB

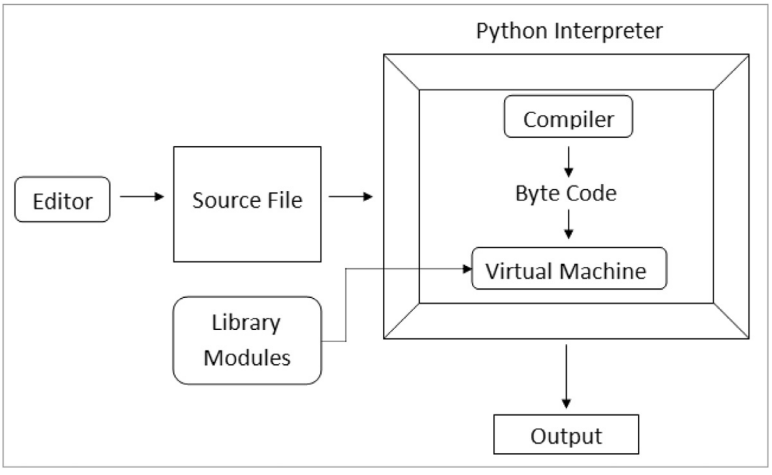


Fig. 4. System Design.

will not result in noticeable human differences because the change is very small. In this method, the bits of the pixels are used. Hence it is necessary to use a lossless compression format like PNG, or the embedded information will be lost in the transformation.

Let 3 pixels are given below in the form 2D matrix; each pixel is of 24 bits.

(01,100,111 11,100,001 11,001,100)
(00,101,111 11,000,000 10,101,001)
(11,010,000 00,100,001 11,111,001)

Character 'A', binary value equals 10,000,001, is inserted into the 2D matrix of the 3 pixels, the following grid results:

(01,100,11**1** 11,100,00**0** 11,001,10**0**)
(00,101,11**0** 11,000,00**0** 10,101,00**0**)
(11,010,00**0** 00,100,00**1** 11,111,00***0***)

All the bits of 'A' are stored in the LSB of each color, but LSB of the third color of the third pixel is made 0, it acts as a delimiter, which signifies the end of the message data. Only four bits are changed out of 72 bits to insert the character successfully in the above case. On average, less than half of the bits are required alteration to embed the message in the cover image. The modification made to the LSB is too small to be noticed by the human eye, so the message is hidden efficiently.

For Example, Text=" Steganography" is embedded in the below cover image:

Fig. 5. Cover Image.

Several mechanisms exist for disseminating information in various media. LSB coding is used to manipulate images or compression technologies that affect image relations. Basic replacement systems attempt to encode hidden information by substituting inconsequential portions of the cover with secret message bits. If the receiver understands where the sensitive information is embedded, he can extract it. A passive attacker is unaware of the modest alterations made throughout the embedding process.

Incorporating the encryption algorithm in further, it enhances the security of the proposed mechanism.By using the SKC (Symmetric Key Cryptography) technique, the watermark is encrypted, which protects against threats. The two different algorithms proposed for image encryption and decryption (Table 3).

At first, we have to add the plain text which is to be encrypted then we have to use the encryption algorithm (AES). So, that it converts the plain text to the ciphertext, now we have to decrypt the message so again we have to use the decryption algorithm so that it can convert the ciphertext into the plain text again and then send it to the receiver.In this context,confidentiality and data integrity are the important and its like as a safefuard against the unauthorised access and use.The two essential things for data sequrities are cryptography and steganography. The cryptography providesconceals the message existence and other technique is allowed to modity the contents . The data is encrypted and sent after being converted into some other form of gibberish using cryptography .In steganography, a picture file is delivered with data contained inside. This study emphasises how combining steganography and encryption techniques can improve the security of communication The above-mentioned (Figs. 7-9) means encryption and decryption techniques which are implemented and demonstrated. Using multiple layers of LSB embedding, such as multi-bit LSB, increases the security of the hidden data. However, it also makes the steganography more detectable and can reduce the quality of the host image. Our analysis shows a balance between security and image quality must be maintained (Figs. 5–9).

Fig. 6. Stego Image.

Table 3
Proposed algorithm for watermarking.

Proposed algorithm for watermarking
The required input: We will provide the grayscale image
1: Take the gray picture W.
2: For every pixel (Pi) of W, produce the decimal value.
3: Determine the accurate binary representation for each Pi..
4: Invert the binary number Bv of 8 digits to get Rv.
5: Find the Key (Ke) of a four-digit divisor.
6. Next divide the reversed Rv number with the Ke divider.
7: The rest and quotient are stored next in an 8-bit array. If necessary, add zeros to the left of the remaining bits and quotient bits to complete an 8-bit string.
Output: **The proposed algorithm produces encrypted data (ED).**

Fig. 7. Steganography Encrypt Image selection.

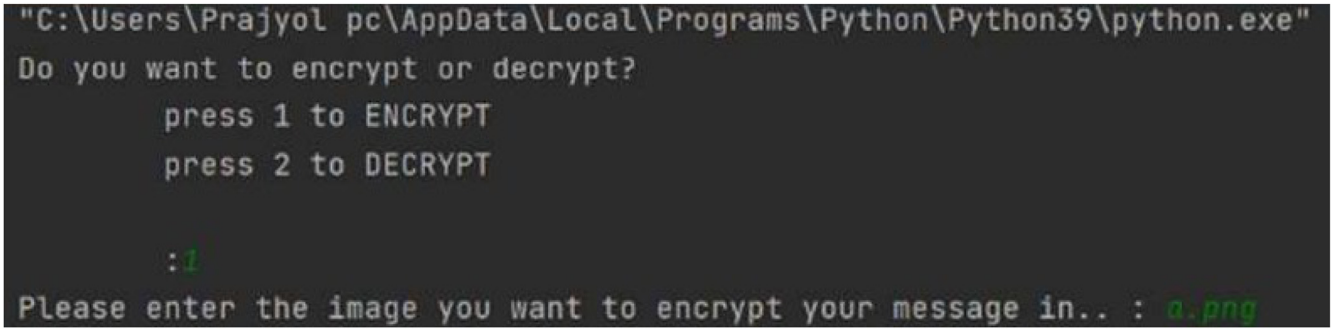


Fig. 8. Message selection.

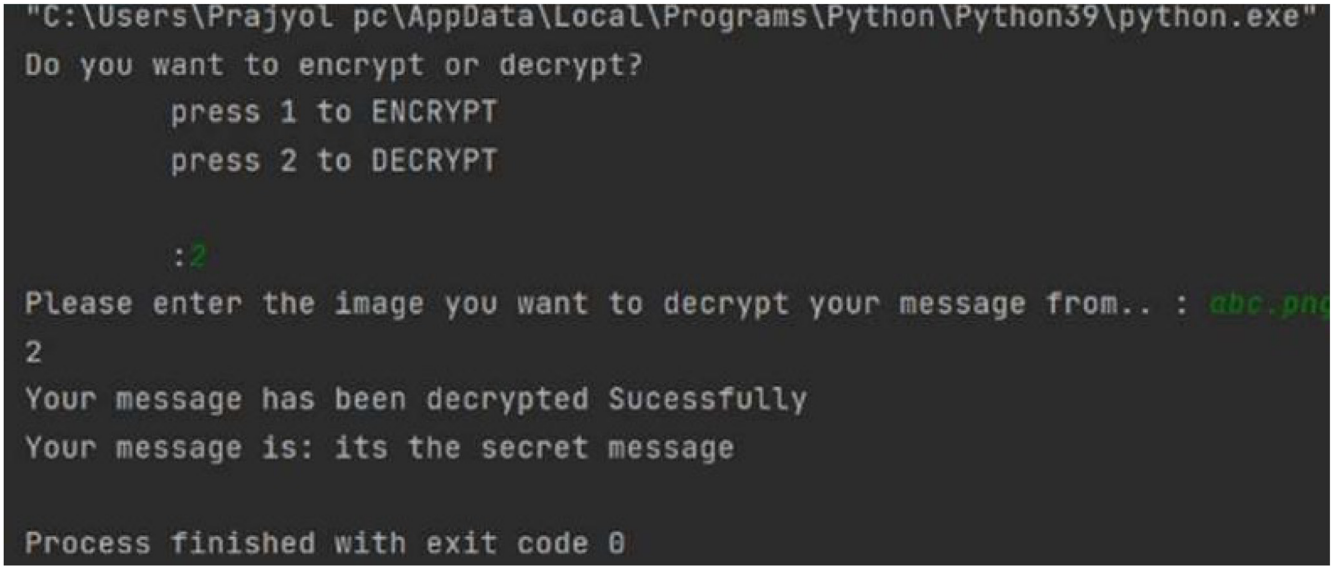


Fig. 9. Decrypted message.

The above-mentioned Tables 4 and 5 are represented for proposed watermarking and decrypted watermarking algorithms

8. System design

8.1. Use case diagram

A Use Case Diagram is a representation of user interaction with the system. When retrieving, the user selects the stego image after initially selecting the cover image, which is transformed into a stegno image. Then, decryption is applied to the segno image. From the cover image to the stegno image, it has undergone several steps. First of all, the cover image generates the ciphertext with the help of a secret text message. Then the ciphertext used the LSB technique to convert into a stegno image. On the receiver side, the user needs to select the stegno image, then after the novel, the LSB decryption technique is used to transform it into the ciphertext to get the secret text message. The above Fig. 10 is used for text hiding using a use case diagram.

8.2. Invisible watermarking

The user selects the cover image at the beginning, then from the cover image generates the text data to produce the QR code. The QR code is used for invisible watermark implementation purposes. Finally,

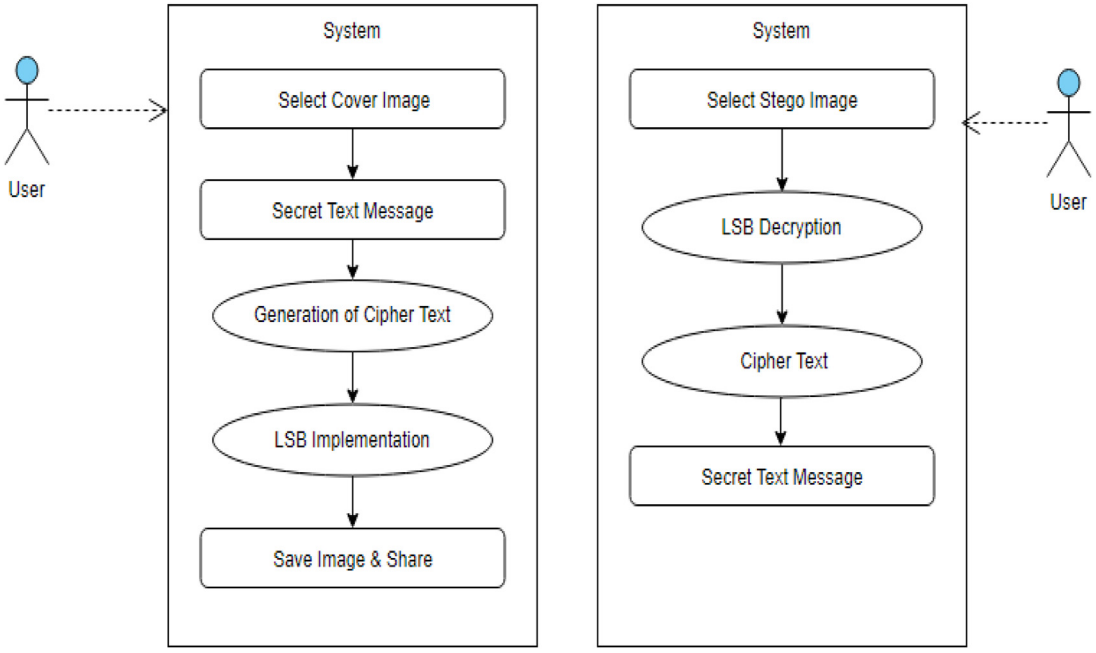


Fig. 10. Use case diagram for Text Hiding.

Table 4
Proposed algorithm for decryption watermarking.

Proposed algorithm for message extraction
The objective of this algorithm is to input the stegno image as well as the secret key (stegno-key), and our implementation part shows the hidden message. Thus, this proposed algorithm is called a message extraction algorithm. 1. Open the image file Stego in reading mode, and read the RGB from the image file Every pixel color. 2. Extract the host images which contain the Red component. 3. Read every pixel's last bit. 4. Initialize the random key, which gives the message bits location in the red Pixels, which are randomly inserted. 5. Select the pixels for decoding and Extract the red pixels LSB value. **Output**: **At the end of this algorithm, our system will provide the secret message behind the stegno image**

Table 5
Proposed algorithm for message extraction.

Proposed Algorithm for Decryption Watermark
Input: Take the two parameters: The Encrypted image data (ED) along with the encryption key (Ke) 1: To generate F, multiply the key (Ke) by the quotient bits of encrypted data (ED). 2: To get G, add the remaining bits of the encrypted data (ED) with the result produced in step (F) above. 3: When the result generated (G) is not an 8-bit number in the previous step, i.e., step 2, we need it to make it an 8-bit number. 4: In order to obtain the decrypted data (DD), flip the G number. Output After executing, we will get the decrypted image

it converts into a hidden stegno image. The other end receives the stegno image in the form of watermarked; then, the extraction algorithm generates the QR code. Finally, scan the generated QR code. The entire process is called an invisible watermarking technique. The above Fig. 11 is used for the Invisible Watermarking use case diagram.

8.3. Visible watermarking

The below-mentioned Fig. 12 is used for Visible Watermarking. In this case, the user submits the data as a text or image. We have developed a model that allows the watermark implementation module to extract the image for sharing purposes.

8.4. The embedding algorithm

The embedding uses a series of pseudo-random data or grayscale images as secret data, together with a grayscale cover picture. A grayscale stego image is the end outcome. Let the remainder and quotient arrays be A and B, respectively. The size of each array may not exceed half that of the cover image.

The above algorithm mentioned in Table 6 is the proposed embedding algorithm that is used in our application

9. Iimplementation setup

We employ the feature similarity (FSIM) and peak signal-to-noise ratio (PSNR) metrics to assess the effects of image segmentation. PSNR is used to compute the PSNR of the segmented image and the original image.

The PSNR index is determined using the following formula:

$$PSNR = 20LOG\left(\frac{255}{RMSE}\right)(dB) \tag{1}$$

Where

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{n}\left(I(i,j)-\hat{I}(i,j)\right)^2}{MXN}} \tag{2}$$

The parameters M and N are termed as the image size, and I is the color of the original image

The Feature Similarity (FSIM) method is used to determine how structurally identical the original and segmented images are. FSIM is described as follows:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x).PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{3}$$

The above equation Ω indicates the entire color image and the FSIM can be represented as the RGB color.

$$FSIM = \frac{1}{O}\sum_{o} FSIM\left(x^0, y^0\right) \tag{4}$$

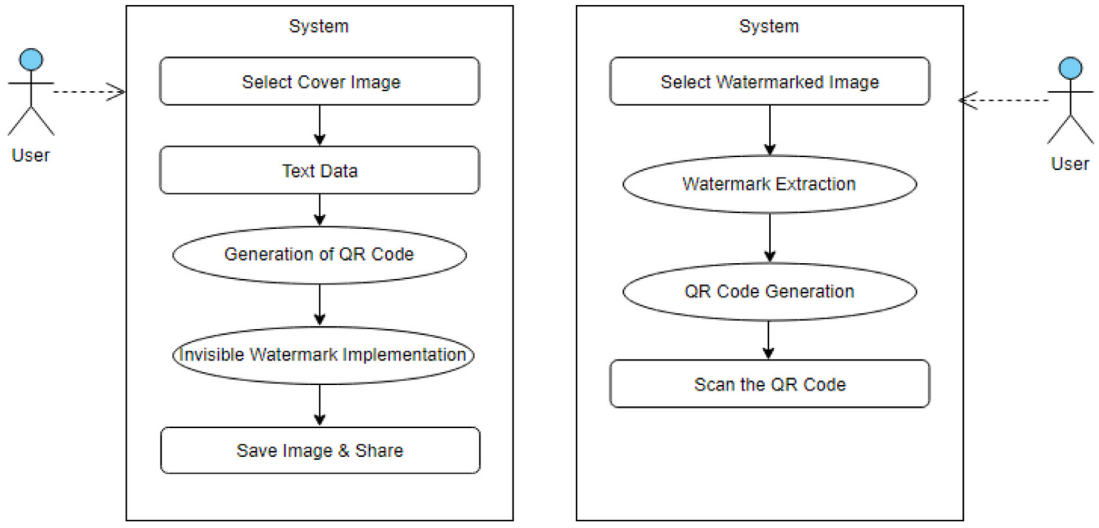


Fig. 11. Use case diagram for Invisible Watermarking.

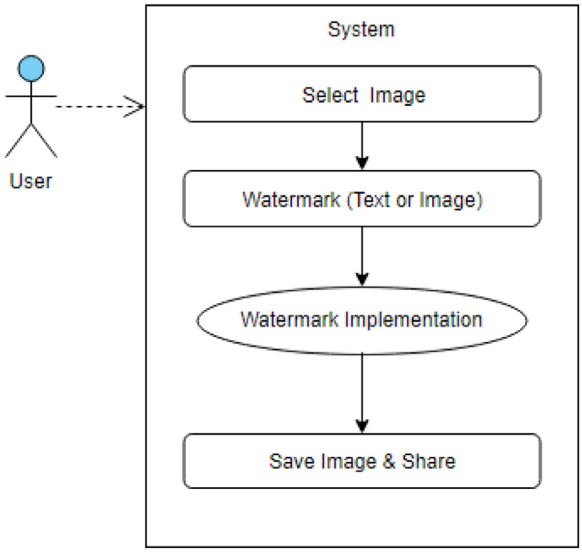


Fig. 12. Use case diagram for visible watermarking.

The above Eqs. (2-4) are used for feature similarity purposes. The term X0, y0 reflect the original image's oth channel and the segmented image's oth channel, where o is the channel amount. Data hiding has an effect on the video bit rate in addition to its effect on the perceptual quality of the video. In general, as compared to the original compressed video, the bit rate of the marked compressed video would increase. To assess performance further, the bit-rate increase ratio (BIR) is introduced, which represents the bit rate created by the proposed data embedding encoder relative to the bit rate generated by the original encoder. Because zero QTCs are not affected during data embedding, the bit-rate increment remains within an acceptable range.

$$BIR = \frac{BR_{em} - BR_{or}}{BR_{or}} \times 100\% \tag{5}$$

$$y_i = \begin{cases} x_{i'}\ if\ m_i = LSB(x_i),\ \ m_{i+1} = f(x_i, x_{i+1}), \\ x_{i'}\ if\ m_i = LSB(x_i),\ m_{i+1} \neq f(x_i, x_{i+1}), \\ x_{i,} - 1,\ if\ m_i \neq LSB(x_i),\ \ m_{i+1} = f(x_i - 1,\ x_{i+1}), \\ x_{i,} + 1, if\ m_i \neq LSB(x_i),\ \ m_{i+1} \neq f(x_i - 1,\ x_{i+1}), \end{cases} \tag{6}$$

$$y_i = \begin{cases} x_{i+1,}\ if\ m_i = LSB(x_i),\ \ m_{i+1} = f(x_i, x_{i+1}), \\ x_{i+1} \pm 1,\ if\ m_i = LSB(x_i),\ m_{i+1} \neq f(x_i, x_{i+1}), \\ x_{i+1'}\ if\ m_i \neq LSB(x_i),\ \ m_{i+1} = f(x_i - 1,\ x_{i+1}), \\ x_{i+1'}\ if\ m_i \neq LSB(x_i),\ \ m_{i+1} \neq f(x_i - 1,\ x_{i+1}), \end{cases} \tag{7}$$

Table 6
Proposed embedding algorithm.

Proposed embedding algorithm

1. The maximum and minimum values in the secret data are determined using the decimal number between 0 and 255.
2. Subtract the greatest value from the minimum and each value of secret data.
3. After dividing the outcome by 32, store the quotient in array A, and divide the remaining amount by 8.
4. In array B, store the quotient from the second division and the remainder in array C.
5. Divide the greatest value by 32 and 8 and save the values in the first pixel of A, B, and C, correspondingly.
6. By inverting the values of these five LSBs, you can embed quotients of the maximum value in the first five LSBs of the cover image's first half.
   a) If B(1, 1) = 1, then flip the value of the first LSB.
   b) Reverse the second LSB if B(1, 1) equals 2.
   c) Reverse the first and second LSBs if B(1, 1) equals 3.
   d) Reverse the third LSB if A(1, 1) equals 1.
   e) Reverse the fourth LSB if A(1, 1) equals 2.
   f) Reverse the third and fourth LSBs if A(1, 1) equals 3. Invert the value of the fifth LSB if A(1, 1) equals 4.
   g) Invert the values of the third and fifth LSBs if A(1, 1) equals 5.
   h) Invert the values of the fourth and fifth LSBs if A(1, 1) equals 6.
   i) Invert the value of the third, fourth, and fifth LSBs if A(1, 1) equals 7.
7. As mentioned in the previous step, embed the remainder of the highest value in three LSBs of the first pixel in the second half of the cover image.
8. Use the fourth bit of this pixel's value as an indicator for R. If this value is less than or equal to 127, the four LSBs of each pixel are inverted in the first step of the embedding process, otherwise, five LSBs are inverted. Because dividing R values in the range of 0 to 127 by 32 requires two bits for representation while dividing R values above 127 requires three bits.
9. Apply the best LSBs algorithm to the pixels you got in steps 5 and 7.
10. To embed each pixel in arrays A, B, and C in pixels of the cover picture, repeat steps 5 and 6.
11. For each pixel in the first half, invert certain inverted bits once more. One or two inverted bits will be inverted again in one of two circumstances. The value of R will determine this.
12. After using the optimal LSBs approach, each pixel now has two values; choose the one that is closest to the original value and return 0 or 1 as an indicator of which invert was chosen.
13. According to this indicator, invert the fourth bit of the pixel in the second portion.
14. Apply the best LSB technique to the pixel from the previous step.
15. The stego picture is formed after embedding everything in the cover image, and it is then sent to the receiver.

Table 7
Characters of the two approaches are compared.

Techinques	Imperceptibility	Robustness	Capacity	Tamper Resistance
Simple LSB	H*	L	H	L
Pseudo-Random Encoding	HH*	L	H	H**

The following terminology is used in the above-mentioned table and represented below as follows:
H: High.
L: Low.
HH: Higher.
The single (*) represents depending on the cover image utilized and the double(**) used the key as well as the random seed. During our test, we used a grayscale/RGB image as the cover image.

Table 8
Pseudo-random encodings PSNR.

Cover Image	Secret Message	Stego-Image	SNR(dB)	MSE	PSNR(dB)
Gray image	Text message	Gray image	69.5043	0.0563	71.4733
RBG image	Text message	sisbr	71.3649	0.031	77.6697
RBG image	Image	Hydrang	63.9812	0.0912	68.5311

Table 9
Least significant bits encoding PSNR.

Aspect	SNR (dB)	MSE	PSNR (dB)
Gray image, Text message	59.5022	0.0673	71.4833
RGB image, Text message	71.4659	0.0312	77.7687
RGB image, Image	63.9921	0.0922	68.5321

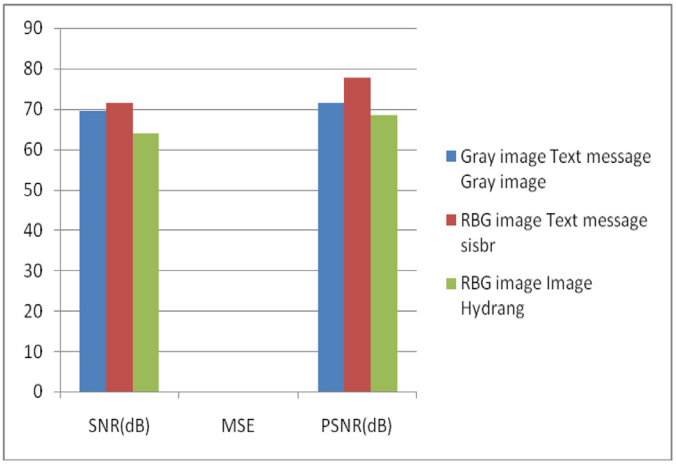


Fig. 13. Comparision of pseudo-random encodings PSNR.

$$Y_{i+1} = \begin{cases} x_{i+1'\, if m_{i+1}} = f\left(x_i . x_{i+1}\right), \\ x_{i+1} \pm 1, \ else \end{cases} \tag{8}$$

To get the message, we only need to get mi from the first pixel's LSB and mi+1 from the embedding phase's function.

In LSB MR, each of the message's two bits is hidden in the brightness of two continuous pixels with just one change. This approach can only be used to hide two bits. We expand LSB MR to hide three bits with only one chance in this paper. As the number of modifications decreases, the PSNR metric rises, strengthening our method's defenses against attacks to reveal the secret message. The LSB-based methods implement their changes at random, without taking into account the image's statistics. In [17], Huang and Ouyang suggested a method for selecting acceptable embedding places in LSB. Smooth image regions, redundant image regions, and fragile image regions were the names given to the picture areas. A small alteration in the image's weak sections makes the image identifiable, and the authors proposed that vulnerable regions be protected by avoiding embedding messages within them.

The above two strategies will be implemented and evaluated (Tables 7 and 8).

We built the aforementioned two ways in MATLAB, and the above-mentioned picture steganography algorithms have their own set of strengths and weaknesses. As a result, deciding on the most appropriate strategy to use is critical. As previously stated, numerous parameters can be used to assess the effectiveness of the steganographic system. The following are some parameters [18]:

Perceptibility is the degree to which embedding information distorts the cover medium to an unacceptably high degree.The Capacity refers to the quantity of information that can be concealed within a given medium, considering the perceptibility of changes, over a specified duration.stego medium in an attempt to delete, erase, or change the embedded data

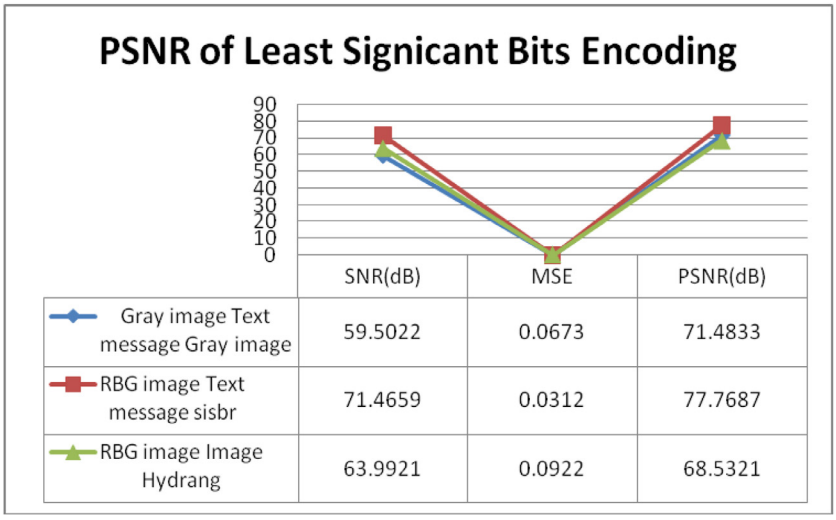
The above-mentioned table 9 is meant for performance metrics for the least significant bits encoding PSNR.It describes the summarizing the PSNR values and their performance metrics for our proposed model.

The above Figs. 13 and 14 are used for comparison between the Least Significant Bits Encoding PSNR and Pseudo-Random Encodings PSNR techniques. The Fig. 15 depicts the learning curves for MSE and PSNR. The blue line indicates the MSE followed by the green line denotes as the PSNR in dB.

The above mentioned Fig. 16 is SNR curve that shows that how SNR values are different for combination of images as well as secret image (Figs. 17–22).

The second experiment will determine how well the suggested steganography technique is at concealing information in speech signals from diverse languages. Finally, Google Translate is used to translate the text to be spoken into various languages.

In the above implementation, these are the following facts found. Images can be altered by nosing, diluting, contrast changes, etc. There should be more than enough pixels of information to insert into the image. There is a possibility that the message will be exposed while someone is screaming; if more than two people have the steganography app, the secret message may also be discovered. If someone sees the code or has access to the software, then they can manipulate the data to their discretion. Reverse engineering can be applied because the application is written in Python, which is an open-source language. Only intended users must know about the working of this software to keep confidentiality. An intruder can suspect images by noise analysis and by some other parameter—protection from Data Alteration by the third party who is constantly searching for potential customers. In practical use, embedded data are weak in most steganography application software. This brings up new possibilities for an information-alteration



	SNR(dB)	MSE	PSNR(dB)
Gray image Text message Gray image	59.5022	0.0673	71.4833
RBG image Text message sisbr	71.4659	0.0312	77.7687
RBG image Image Hydrang	63.9921	0.0922	68.5321

Fig. 14. Comparision of least significant bits encoding PSNR.

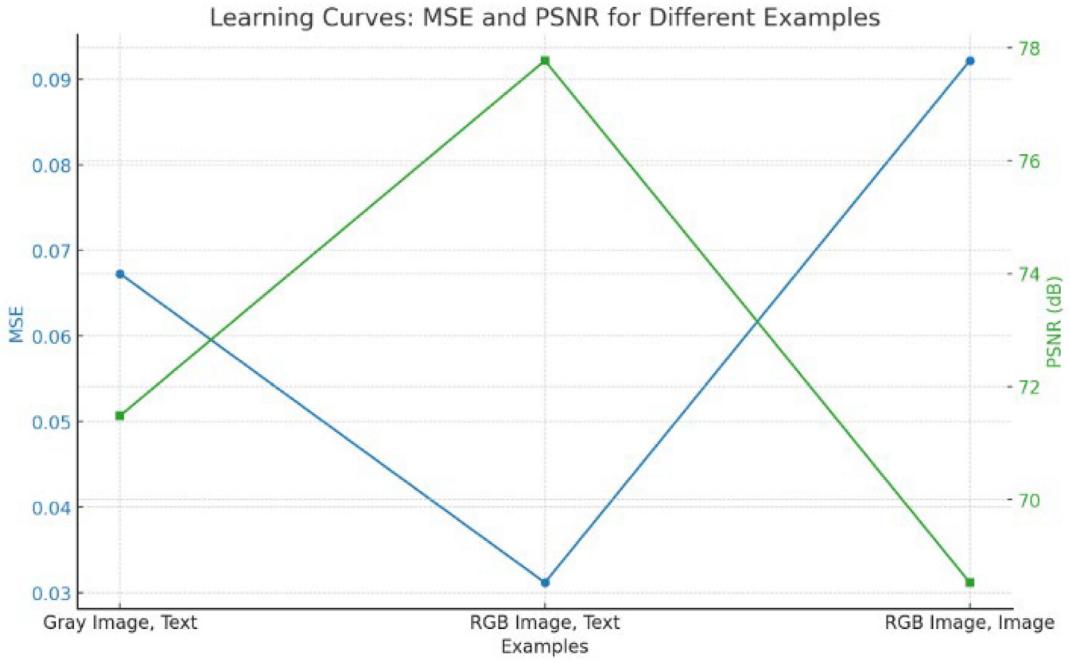


Fig. 15. Learning curves for MSE and PSNR.

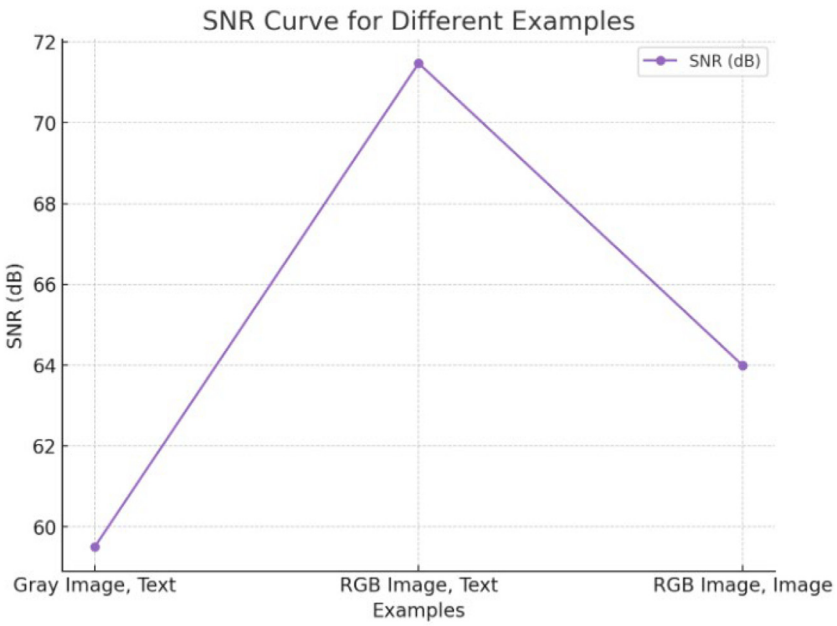
protection application, such as a Digital Certificate Document System. People can transfer their digital certificate data to any person in the globe without risk of change, forgery, or tampering. Access Control System because we choose with whom we have to share our information. Based on the current situation, digital content is increasingly circulating, and we can cover our information for fair use by using the access control mechanism by supplying an access key to extract information. E-commerce for the life insurance and banking sectors and Media, which needs to be encrypted for data breaches and dealing with the customer's privacy, is a severe concern of the institution, so here steganography comes in handy. Digital Watermarking, which may be visible or invisible to the human eye, provides authentication for the originality of the documents such as images and video. The reason behind using LSB steganography is to efficiently embed the data with minimal distortion as a comparison to other techniques like DCT or DWT.LSB provides a good balance of simplicity as well as computational efficiency of our application.

9.1. Hiding the message and Extract the hidden message.

These are the following steps that we have done in our implementation.

STEP 1: In the above implementation, we conceal our secret message and will make assured that it is properly encrypted so that someone would want to extract the message from the image, he would not be able to decipher the secret message we have hidden. In order to construct our secret key, we will first start nano by typing nano file.txt, then pressing return, and then we will enter our secret key before exiting the program.

STEP 2: When we send our message, we'll use the symmetric key algorithm (as implemented by gpg), which will allow both the sender and the recipient to use a single key for both encryption and decryption.



Figs. 16. SNR Curve.

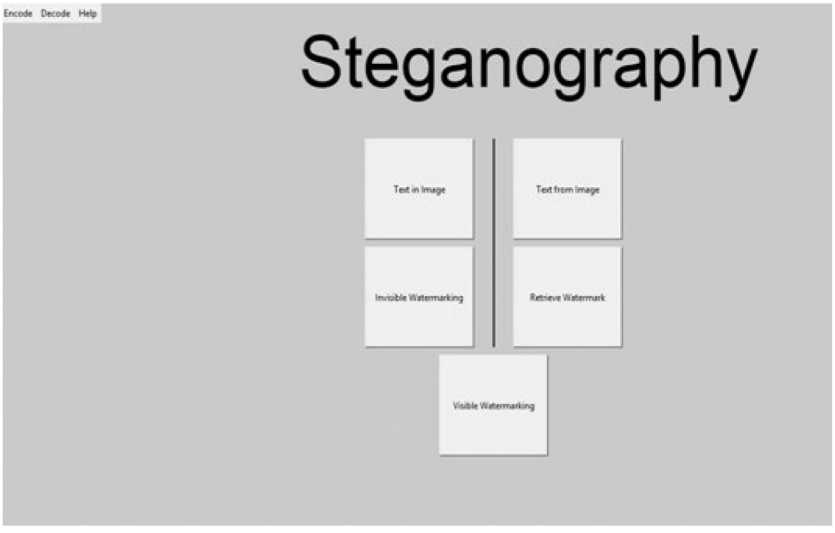


Fig. 17. Main menu of Steganagraphy.

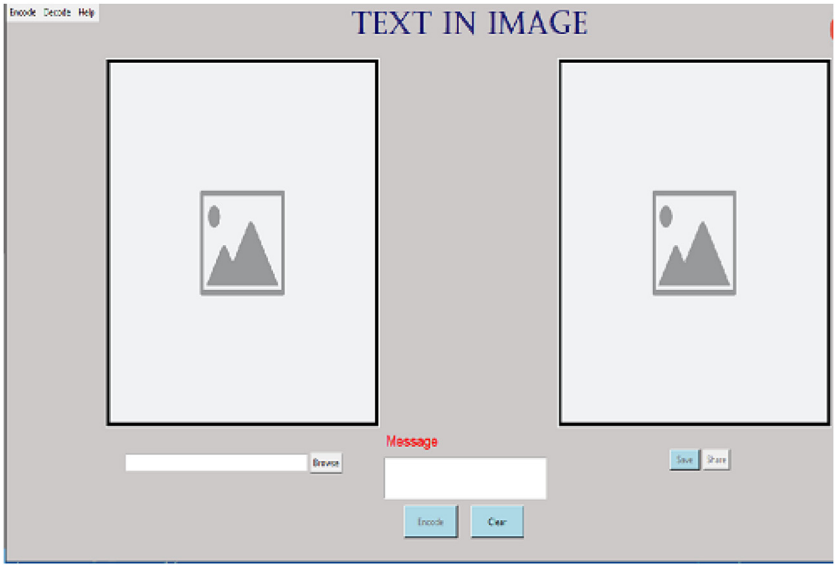


Fig. 18. Text in Image for convertion.

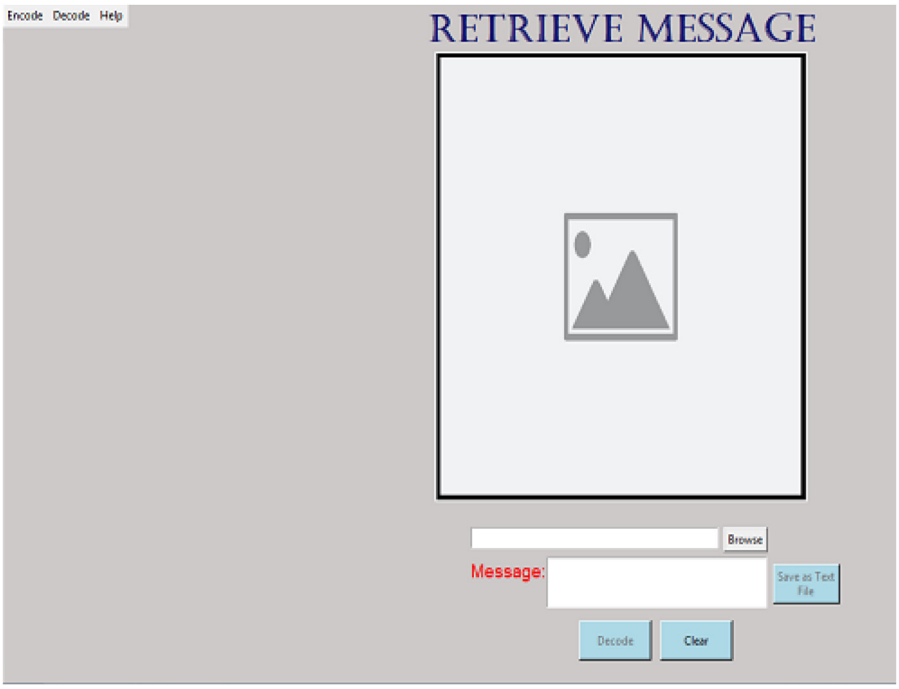


Fig. 19. Retrieve text.

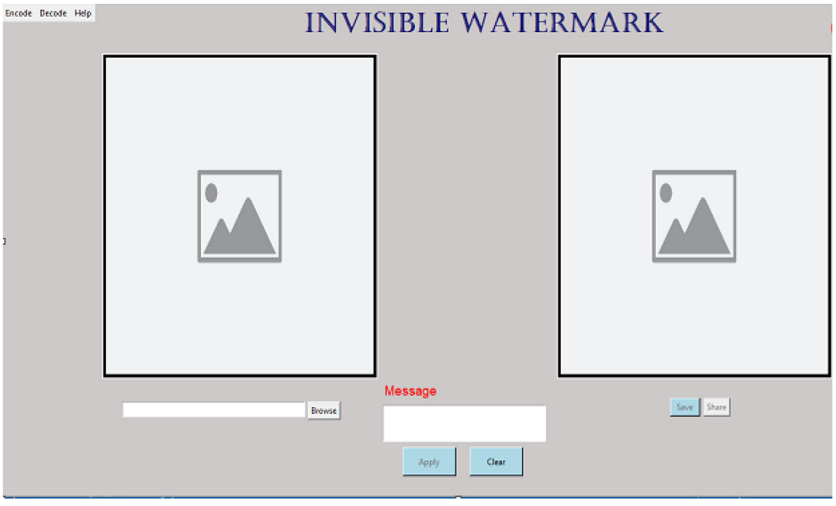


Fig. 20. Invisible watermarking.

type gpg -c msg.txt to encrypt it.

Step #3: It will prompt you for a password; simply type in the one you like.

Step #4: Uploading this encrypted message to a hosting server and then copying the downloading link to include it in our image is how we accomplish this task.

Step#5: To decode the hidden information contained within the image

type decode –i " IMG" -o "Target o/p format"

Step#6: We have successfully extracted the file from the image; now, it is time to read the information contained within it.

type cat msg.txt to read the file.

Step#7: It is now necessary to decrypt the file that has been downloaded. This program will prompt us to enter a password, after which we will receive our secret message.

Identifying and extracting the hidden message Our recipient has successfully received the image file, and it is now time to expose the concealed message contained within it.

The similarity between the original cover speech signal and the stego speech signal that contains the hidden data is referred to as the consistency attribute. The Mean Absolute Error (MAE) and the Mean Square Error (MSE) can be used to compare the similarity of two signals (MSE). In addition, a new metric, the Segmental Signal-to-Noise-Ratio, is used in this paper to assess the accuracy of the stego signal (SNRSeg). The SNRSeg is a favored speech quality indicator that yielded more consistent results than traditional quality indicators like the MSE and MAE. The SNRSeg calculates the average of short segment Signal Noise Ratio (SNR) values. Calculating the difference between the two test signals sample by sample yields the SNR.

$$SNRS_{eg} = \frac{10}{M} \sum_{m=0}^{M-1} log_{10} \sum_{i=Nm}^{Nm+N-1} \left(\frac{\sum_{i=1}^{N} x^2(i)}{\sum_{i=1}^{N} (x(i) - y(i))^2} \right) \quad (9)$$

The terms M and N are the number of segments and the length of segments, respectively.The SNRSeg has produced more consistent results than the standard SNR metric, especially for waveform encoders.

This section is primarily used for the proposed methodology based on the extraction process. It can also be possible to extract the signals when

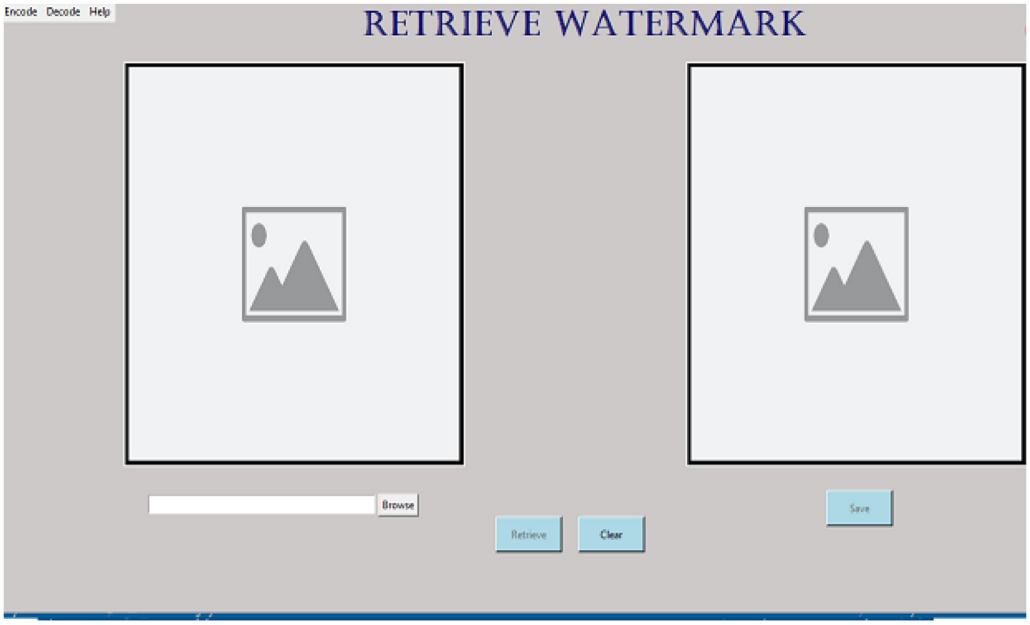


Fig. 21. Retrieve watermarking.

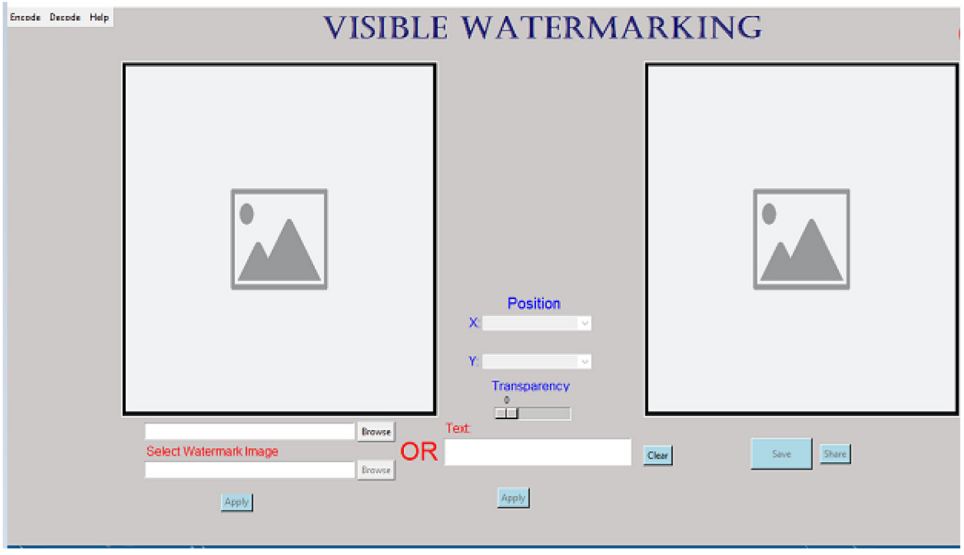


Fig. 22. Visible watermarking.

we provide the secret speech. We need to provide samples of speeches to retrieve similar kinds of embedding types.

Steps

The same splitting threshold value used at the embedder is used to segment the obtained stego speech signal.

- √ Each section is subjected to the DCT.
- √ Since the magnitude contains the hidden data, only the magnitude of the DCT is extracted.
- √ To locate the hidden data, each segment is subjected to the 64-bit quantization

10. Conclusion

It is observed that the results obtained in data hiding are very impressive through the LSB steganographic substitution process because it uses the simple fact that each image can be broken up into individual bitplanes, each consisting of different information levels. However, it should be noted that the approach is only useful for JPG and PNG images as these require lossless compression techniques, as discussed above. It is also important to note that while steganography was once undetected, with the various methods currently being used, it is easier to detect the presence and easier to obtain.

Future scope

Steganography could allow for avoiding inaccuracies in the printing and scanning of media(data).: Can steganography be detected in image files? That is a severe issue. A simple steganographic technique can be identified by merely analyzing the low-order bits of the bytes in the image. However, if the steganographic algorithm is more complicated and the embedded data is scattered over the image, this is random or encrypts the information before embedding. Inline images are widely used by the World Wide Web (www). Globally, there are millions of pictures on different websites. An application can be created to act as a

web browser to retrieve images embedded in the web page. This stego-web will work on top of the current Website and could be a way of transmitting information secretly. The proposed embedded algorithm is designed for gray-scale images for secret data in the embedding. Our application developed this proposed algorithm. In conclusion, the proposed LSB steganographic technique demonstrates high effectiveness in hiding image data, as evidenced by the presented PSNR values. This method can be effectively utilized in secure communication applications, contributing to advancements in the field of cybersecurity and digital data protection.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Rasmita Panigrahi: Writing – original draft, Methodology, Conceptualization. **Neelamadhab Padhy:** Writing – review & editing, Methodology, Investigation, Formal analysis.

References

[1] R.W. Alma, Wardhani, S. Muhasyah, M Delina, The least significant bit of steganography method for digital data protection in the barcode, AIP conference proceedings, 2169, AIP Publishing LLC, 2019.
[2] M. Ramesh, G. Prabakaran, R. Bhavani, QR-code image steganography, in: Proceedings of the second international conference on emerging research in computing, information, communication, and applications, 2014.
[3] B. Chitradevi, N. Thinaharan, M. Vasanthi, Data hiding using the least significant bit steganography in digital images, Statistic. Approach. Multidisciplin. Res. 1 (2017) 144–150.
[4] J. Zhu, R. Kaplan, J. Johnson, L. Fei-Fei, Hidden: hiding data with deep networks, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 657–672.
[5] S.N. Bal, M.R. Nayak, S.K. Sarkar, On the implementation of a secured watermarking mechanism based on cryptography and bit pair matching, Journal of King Saud University-computer and information sciences, 2018.
[6] K.H. Jung, K.Y. Yoo, A steganographic method based on interpolation and LSB substitution of digital images, Multimed. Tools. Appl. 74 (6) (2015) 2143–2155.
[7] Bamatraf, A., Ibrahim, R., Salleh, M., & Mohd, N. (2011). A new digital watermarking algorithm using a combination of least significant bit (LSB) and inverse bit. arXiv preprint arXiv:1111.6727.
[8] X. Duan, D. Guo, N. Liu, B. Li, M. Gou, C. Qin, A new high capacity image steganography method combined with image elliptic curve cryptography and deep neural network, IEEe Access. 8 (2020) 25777–25788.
[9] J.L. Pichardo-Méndez, L. Palacios-Luengas, R.F. Martínez-González, O. Jiménez-Ramírez, R. Vázquez-Medina, LSB pseudorandom algorithm for image steganography using skew tent map, Arab. J. Sci. Eng. (2019) 1–20.
[10] Y. Zhang, C. Qin, W. Zhang, F. Liu, X. Luo, On the fault-tolerant performance for a class of robust image steganography, Signal. Process. 146 (2018) 99–111.
[11] G. Swain, Very high capacity image steganography technique using quotient value differencing and LSB substitution, Arab. J. Sci. Eng. 44 (4) (2019) 2995–3004.
[12] A. Nolkha, S. Kumar, V.S. Dhaka, Image steganography using lsb substitution: a comparative analysis on different color models, in: smart systems and iot: Innovations in computing, Springer, Singapore, 2020, pp. 711–718.
[13] M. Baziyad, T. Rabie, I. Kamel, Achieving stronger compaction for dct-based steganography: a region-growing approach, in: World conference on information systems and technologies, Cham, Springer, 2020, pp. 251–261.
[14] R. Sridevi, A. Damodaram, S.V.L. Narasimham, Efficient method of audio steganography by modified lsb algorithm and strong encryption key with enhanced security, J. Theoret. Appl. Informat. Techn. 5 (6) (2009).
[15] J. Liu, K. Zhou, H. Tian, Least-significant-digit steganography in low bit-rate speech, in: 2012 IEEE International Conference on Communications (ICC), IEEE, 2012, pp. 1133–1137.
[16] M.A. Ahmed, M.L.M. Kiah, B.B. Zaidan, A.A. Zaidan, A novel embedding method to increase capacity and robustness of low-bit encoding audio steganography technique using noise gate software logic algorithm, J. Appl. Sci. 10 (1) (2010) 59–64.
[17] Q. Huang, W. Ouyang, Protect fragile regions in steganography LSB embedding, in: Proceedings of the 2010 3rd International symposium on knowledge acquisition and modeling (KAM), Wuhan, China, 2010, pp. 175–178.
[18] A review of data hiding in digital images by E Lin, E Delp center for education and research information assurance and security purdue university, west lafayette, IN 47907-2086.