



An enhanced Huffman-PSO based image optimization algorithm for image steganography

Neha Sharma¹ · Usha Batra¹

Received: 24 May 2019 / Revised: 5 December 2020 / Accepted: 15 December 2020 /
Published online: 1 January 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

It is crucial in the field of image steganography to find an algorithm for hiding information by using various combinations of compression techniques. The primary factors in this research are maximizing the capacity and improving the quality of the image. The image quality cannot be compromised up to a certain level as it breaks the concept of steganography by getting distorted visibly. The second primary factor is maximizing the data-carrying/embedding capacity, which makes the use of this technique more efficient. In this paper, we are proposing an image steganography tool by using Huffman Encoding and Particle Swarm Optimization, which will improve the performance of the information hiding scheme and improve overall efficiency. The combinational technique of Huffman PSO not only offers higher information embedment capabilities but also maintains the image quality. The experimental analysis and results on cover images along with different sizes of secret messages validate that the proposed HPSO scheme has superior results using parameters Peak-Signal-to-Noise-Ratio, Mean Square Error, Bit Error Rate, and Structural Similarity Index. It is also robust against statistical attacks.

Keywords PSO · Huffman-PSO · DWT · Image steganography

1 Introduction

The advancement in the field of scientific computation areas and increasing average internet usage per person due to better availability of devices have enabled us to create new and better opportunities for enhancing the levels of noiselessness and improved secrecy of communication such that steganography has advanced to a level of providing better secure algorithms for image embedding etc. The rapidly increasing internet users across the platforms and sectors have increased the potential of information hacking, information tempering, and loss of information.

✉ Neha Sharma
nehasharma0110@gmail.com

¹ GD Goenka University, Gurugram 122103, India

These issues become more critical, especially when information is either very sensitive/critical or confidential. Steganography offers one such solution to avoid any potent risk/attack for data and minimizing its potent possible risk factor [1]. Steganography, a technique that hides the images' information, initially started as hiding texts only, has now advanced by hiding any form of information/multi-media like audio, video, and image files [2]. Steganography aims to enhance the images' capacity for hiding the critical messages or data in the form of image/text/audio/video by maintaining the overall process's efficiency. The critical fact is that it's very tough for any person, even for an intruder, to identify that there is data embedded or hidden in the image as steganography retains the image's original quality [3]. Steganography veils the information in the digitally formatted media files with a principle to veil the information and the details of the transmission mechanism [1]. However, out of all the types of media, images are preferred [4] because they are easy to find and have good redundancy.

Images used for steganography are called the original image and a steganographic image. These are used for data embedding, transferring and data retrieving [5]. Image steganography's objective is to increase the embedding capacity [6] and minimize steganographic image detectability. There are three crucial parameters in the steganographic system that have to be considered the capacity of the images/source, noiselessness of the image/media format, and the robustness of the final output media. Capacity is crucial as it is the extent of the secret data/information implanted in the original object (original image in which the data is being embedded) [7]. Imperceptibility/Noiselessness is the distortion in the image quality after the data has been embedded, which determined whether the original image with the hidden message can be easily marked or distinguished by any professional/spy/hacker. Robustness is the toughness of the image/media source that defined the degree of protection from manipulation during the information transmission. Image steganography can be defined into two main categories (i) Spatial Domain (ii) Transform Domain steganography. Both methods differ in the process of data embedding from each other. Both have a different mechanism of embedding the secret message in the cover image. In Spatial Domain, pixels are moved or shuffled to embed the covert information/data/message, while in the Transform domain, the transform coefficient is used for embedding the secret message [8].

There are both advantages and shortcomings in these two methods. Frequency domain mechanisms are capable of giving better immunity against threats. On the other hand, it lags in concealing higher information against spatial domain mechanisms that can hold more data, but security is dubious [9]. There are many steganography techniques based on transform like Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) [10], or Discrete Wavelet Transform (DWT) [11, 12]. Swarm Intelligence technique is a branch of A.I (Artificial Intelligence). The idea of swarm techniques has been derived from observatory analysis of natural living beings and their routine systems like how ants manage their lives in their colonies, how birds live and stay in their flocks, and how animals tend to behave in herds. Recording and analyzing effectively, such events of behaviors were applied to computationally intelligent systems [13].

1.1 Research motivation

Techniques of steganography, such as image steganography, have received very little attention in information security. More research has been done in the field of cryptography and watermarking. This research identifies three significant challenges in image steganography—robustness, imperceptibility, and high payload. This paper tries to address those challenges. Further, this research aims to identify the most relevant position for embedding the payload through an optimization algorithm and validate the output through the performance metrics defined in the paper. Additionally, this paper also tries to use the Huffman encoding technique to compress and encrypt the secret data to address the gap of high payload.

The paper is organized as the Introduction of the paper, done in Sect. 1. A literature survey of the existing methods is done in Sect. 2, and their advantages and disadvantages are listed for further analysis. The proposed method and its algorithm are discussed in detail in Sect. 3. Later there is a brief discussion that has been done in Sect. 4, and the results are concluded in Sect. 5.

2 Literature review

In this section, the literature review of the existing works has been done. Nipanikar et al. [14] have come up with a method using sparse representation and an algorithm—particle swarm optimization (PSO) to effectively select the pixels to mask secret audio signals in the image using steganography. In the PSO-based mechanism, an algorithm that selects the media file's pixels uses the fitness function, a *prima facie* cost-dependent function. In the data and responses recorded using proposed methods and other techniques, the proposed method has achieved better PSNR and MSE values of 47.6 dB and 0.75. Ziyad et al. [13] has comparatively analyzed the existing standard particle swarm optimization technique (SPSO) with human-based particle swarm optimization (HPSO). His analysis reveals that the performance of their proposed image steganography with HPSO is better than SPSO. They have used human-based steganography to find the best location in the image. The results also reveal the fact that HPSO's time requirements is better as compared to SPSO for the purposes of data veiling and unveiling. Miri el Attell [15] researched out an approach for data veiling in frequency domains using certain genetically mutated algorithms, which he has also proposed in [15]. The authors have used techniques principled on K&C (Kieu & Chang) where the ciphered secret information is veiled in frequency coefficients that predominantly signify the edges of the spatial domains' concealed image. The original image used in the process changes very little, and the resultant image is more suitable for human visualizing and systems. In this entire working process, the frequency's space remains wobbly and gets highly dependent on the type of secret information and the original image. The authors have added extra security layer so that eavesdropper does not know how to find veiled information in the respective space. Their work has shown an improvement of 1.8 dB from their prior work in the PSPNR norm. Rajeswari et al. [16] have implemented steganography with wavelet, particle swarm optimization, and least significant algorithms. The

cover image is processed to 5 levels with the daubauchi-1 wavelet. At the 5th level, the approximation 8×8 matrix is masked in the lower nibble of the original image, further based on the particle's location (x, y) derived by using particle swarm optimization mechanism. The authors have claimed that the proposed algorithm-mechanism performs better than the previous works analyzed during the process. Hamid [17] proposed that his researched & worked out method give better resultant values while veiling a gray scale image behind another gray scale image by DWT for data veiling and a chaotic map for better security concerns.

The author has decomposed the image using db1 wavelet transform and break down the same into 4 sub-bands (LL, LH, HL, and HH). The secret image is then ciphered with logistic chaotic mapping better security concerns. Post the process mentioned here, the ciphered secret image is embedded into the original image's HH band. The analysis of the proposed technique's values is done on the parameters of PSNR correlating the factor values, which are 39.25 dB and 1, respectively. The authors in [18] have recognized the gestures in RGB and RGB-static images using CNN. Their proposed model has achieved a 94.8% recognition rate, and the data set used the ASL dataset. In [19], the authors enhance the quality fingerprint image for 3 reasons: the first being to secure the minutiae template, improve image synthesis, and finally improve the interoperability of template using various algorithms and sensors. The authors in [20] have introduced a simple algorithm to remove the haze. The haze in the images sometimes comes while capturing the image in a bad climate. In order to remove haze, the prior code fading algorithm is used, and then weighted guide image filtering filters the guided filter. Thereby, it makes the scattering coefficient variable and then applies a color balancing algorithm to increase the images' readability. Six meta-heuristics algorithms are used in [21] to automate the process of test suite generation. The performance was evaluated using performance metrics, and it was found that ABC gave optimum results; BA was found to be fast, but results were not optimal. FA was the slowest among all while CS, PSO, and HCA gave an average performance. The authors [22] have used ABC and CSA test suite optimization, and it was found that the average value of path coverage for ABC was 90.6% and that of CSA was 75.4%. ABC algorithm was found to be more reliable.

3 Proposed algorithm

There are many optimization techniques for optimizing images in the field of image steganography. This paper highlights and tries to resolve the gaps that are present in the current literature. The proposed model tries to combine the features of existing algorithms in order to achieve high payload and imperceptibility without compromising the robustness. The decomposition of the image through discrete wavelet transform helps the proposed model in embedding the data. Further, the Huffman algorithm is applied on the decomposed blocks to find the embedding bits. The use of Huffman encoding in the proposed model not only helps in compressing the data but also provides an extra layer of security as it also encodes the data before compression, thereby addressing one of the gaps, i.e., issue of embedding high payload.

The PSO (Particle swarm optimization) algorithm in the proposed model addresses the concern of imperceptibility by identifying the best possible location in the cover image for embedding the secret bits and for the further optimization process. The proposed model also addresses the third gap in the literature, as it is robust against various statistical attacks that can be performed on it. The hybridization of these existing algorithms in the proposed model helps in resolving the gaps to a greater extent as compared to the techniques considered alone. The proposed model is one of the crucial advantages that it uses less storage space for consistently occurring characters. Another significant advantage of the proposed model is that it takes considerably less time to locate the best location for hiding the embedding bits. It is considerably lower as compared to the time taken by the PSO algorithm.

The proposed algorithm is principled on Huffman Encoding and Particle Swarm optimization. It consists of two mechanisms (1) embedding and (2) retrieving the secret message from an image (refer Fig. 3). The implementation of the Huffman-PSO is done using MATLAB. The code works with coloured images. The system architecture of the proposed algorithm is shown in Fig. 1.

3.1 Wavelet decomposition

A wavelet is a waveform that starts at zero, increases, and then decreases, and Wavelet decomposition is to break the signal into scaled and shifted versions of the original wavelet/waveform. The cover image is decomposed into 4 sub-bands (LL, LH,

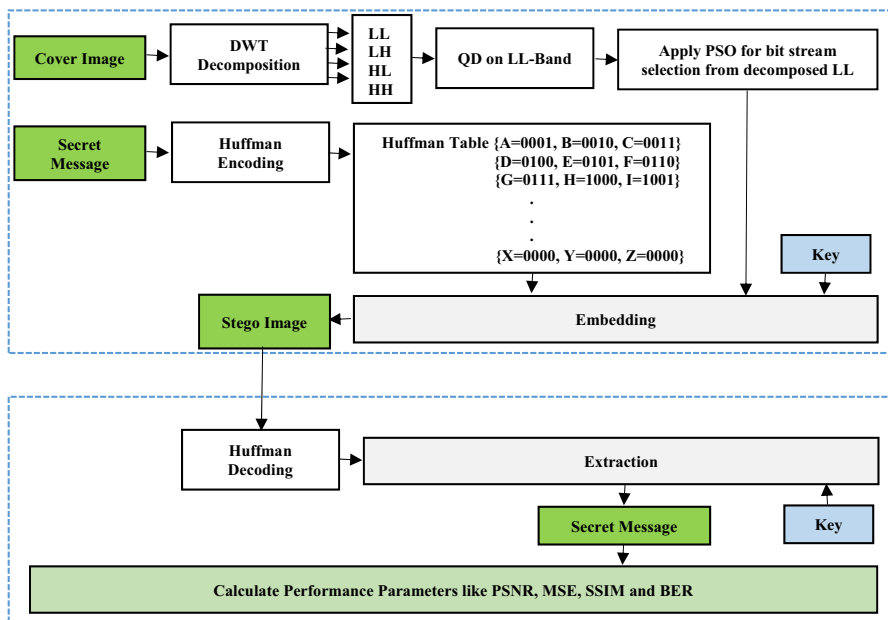


Fig. 1 System architecture of the proposed algorithm

HL, HH) using DWT. Quadtree decomposition is performed on LL band, and then the blocks of division are created to calculate the mean of the decomposed image. The size of the images obtained at each level of decomposition:

Level-1 512X512 original image—> 256×256 (LL) 256×256 (LH) 256×256 (HL) 256×256 (HH).

Level-2 256X256 (Level-1 A)—> 128×128 (LL) 128×128 (LH) 128×128 (HL) 128×128 (HH).

The approximation coefficients in the second level decomposition are used as inputs to the Huffman-PSO algorithm for embedding compressed secret bits in the original image.

3.2 Embedding process

Initially, cover image, secret key, and secret message are attained. DWT is performed on the cover image in order to extract the high and low-frequency wavelets. DWT decomposes the image in 4 sub-bands (LL, LH, HL, HH). Low-frequency wavelet coefficients are selected from the extracted wavelets, and then the optimum pixel is selected to make it fit for data hiding. The PSO algorithm finds the best position to hide the secret data. The secret message is then compressed through the Huffman encoding technique, and the compressed message is then embedded into the decomposed cover image. Finally, in order to create a stego-image, inverse DWT is employed. The embedding process is shown in Fig. 1. To embed the secret message, consider the original cover image as O with dimensions $M \times N$. The image is usually represented with spatial representation. In order to convert it into a frequency domain, DWT is applied. The image then undergoes decomposition and sampling to offer a high degree of robustness. The coefficients thus obtained are:

$$[O_1 \ O_2 \ O_3 \ O_4] = \text{DWT}(O) \quad (1)$$

O_1 is the low-frequency band's coefficient and contains all the significant information of the image; O_2 , O_3 , O_4 are the high-frequency band and contain information like edges of the image. O_1 band is selected for further process and the coefficients extracted are:

$$[O_1^{LL} \ O_1^{LH} \ O_1^{HL} \ O_1^{HH}] = \text{DWT}(O_1) \quad (2)$$

O_1^{LL} is lower frequency band where as O_1^{LH} , O_1^{HL} , O_1^{HH} are higher frequency sub-bands of O_1 . To create stego-image, low-frequency bands are selected by adding the secret text data, represented by D , and the optimal position.

$$C_1^{*j} = C_1^j + (D_i \times P_{opt}) \quad (3)$$

where $j = \text{LL, LH, HL, HH}$, D_i is the secret text data, and P_{opt} is the optimum position to embed the data. To represent the image back after data hiding, inverse DWT is performed.

$$O_1^{**} = IDWT(O_1^{*LL} O_1^{*LH} O_1^{*HL} O_1^{*HH}) \quad (4)$$

The embedded secret data with a modified band is:

$$O^{**} = IDWT(O_1^{**} O_2 O_3 O_4) \quad (5)$$

The algorithm for the same is as follows:

Embedding Algorithm

Input: Original Image, Key (K) & Secret Message

Output: Steganographic Image

Step 1. Apply pre-processing on original image and secret message

- ⇒ Resize original image in MXN
- ⇒ Calculate histogram of original image
- ⇒ Decomposition of cover image using 2-level DWT and decomposed in 4 bands (LL, LH, HL and HH)

Step 2. Apply Quad Tree decomposition on LL to divide image into dimension of [2 64]

Step 3. Create blocks of division to find out the mean of decomposed image

Step 4. Apply Huffman on each block based on Huffman dictionary and find out the pixel range (Embedding Bits) for embedding and initialize the PSO for optimization of pixel range

Step 5. Define swarm size and fitness function of PSO

$$Fit_function_PSO = \begin{cases} True & \text{if } f_s > f_t \\ False & \text{otherwise} \end{cases}$$

Step 6. Apply PSO on Embedding Bits

Calculate size of Embedding Bits (R, C)

For i=1 → R

For j=1 → C

Fs = Embedding Bits (i, j)

Ft = Threshold (i, j)

Fit_function_PSO = Call Fit_function_PSO (Fs, Ft)

No. of Variable = 1

Fitdata = PSO (Fit_function_PSO, No. of Variable, PSO functions)

End

End

Step 7. Check size of original image and confidential message

If original image size > confidential message size

Apply embedding and generate steganographic image using inverse decomposition

Steganographic image = embedding (Original image, Confidential message, key)

Else

Embedding is not possible

End

Step 8. Return: Steganographic Image

Step 9. End

3.3 Secret Key

The secret key in this work is a sequence of integer numbers ranging from {0,1, 2, ...,9}. The secret key is assumed to be pre-shared between the sender and receiver.

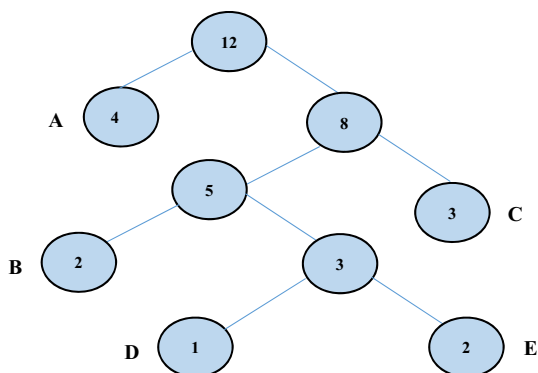
3.4 Huffman Encoding for Secret message

The secret data to be embedded first undergoes Huffman encoding in order to address one of the challenges of steganography, i.e., to embed high payload. Consider the secret message to be ABACEECDABCA. To construct the Huffman table, following steps are followed:

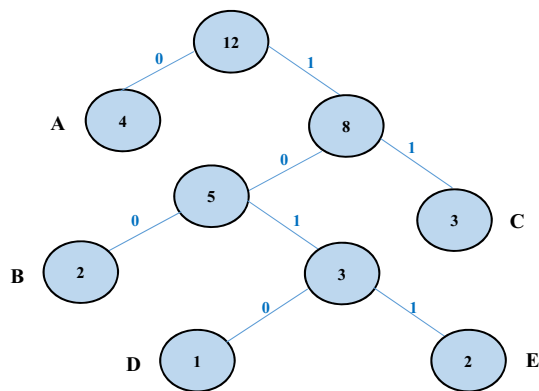
Step 1 The occurrence of each character in the secret message is counted.

Characters	Frequency	Probability
A	4	$\frac{4}{12} = 0.333$
B	2	$\frac{2}{12} = 0.1667$
C	3	$\frac{3}{12} = 0.25$
D	1	$\frac{1}{12} = 0.0833$
E	2	$\frac{2}{12} = 0.1667$

Step 2 As per character frequency, a sorted list is created and a tree is designed.



Step 3 Left of the parent node (12) is marked as 0, and right is marked as 1. This is repeated until the last element is reached.



Step 4 The characters are thus converted or encoded to the following code.

Characters	Code
A	0
B	100
C	11
D	1010
E	1011

The secret text data is converted from ABACEECDABCA to 0,100,011,101,11 0,111,110,100,100,110 and is embedded to the best fit pixel returned by the PSO algorithm.

3.5 Extraction process

The aim here is to extract the secret message from the stego-image successfully. The input in the extraction process is the stego image, which is done on that receiver side. The stego image then undergoes DWT in order to transform the image from spatial representation to frequency representation. The stego image is denoted as O^{**R} . The low and high-frequency coefficients are thus obtained as a result:

$$[O_1^{**R} O_2^{**R} O_3^{**R} O_4^{**R}] = DWT(O^{**R}) \quad (6)$$

The modified band is O_1^{**R} , thus the coefficients of the sub-bands are:

$$[O_1^{LL*} O_1^{LH*} O_1^{HL*} O_1^{HH*}] = DWT(O_1^{**R}) \quad (7)$$

The next step in extraction is to match the secret key; if the key matches, then the original secret message is extracted; otherwise, the encrypted secret message is

retrieved. PSO algorithm is applied in order to obtain the location of hidden data. Huffman decoding technique is applied to extract the secret message. The stego image thus obtained is in binary form. This binary sequence is then converted to character sequence, and hence the original secret message is retrieved. If the secret key pair does not match, then an encrypted image is received as follows:

Encoded Data

10100101011011101101110101001001001010101010101011

Encrypted Image



The encrypted image is generated in order to add another level of security to the proposed model.

Extraction Algorithm
Input: Steganographic Image & Key (K) Output: Secret message & performance parameters Step 1. Read Steganographic image Step 2. Apply DWT on steganographic image Step 3. Match key If Key is CORRECT Apply extraction & recover secret message Secret message = embedding (Steganographic Image, Key) // Original secret message is extracted Else Encrypted image = embedding (Steganographic Image, Key) // Encrypted image is extracted so that it cannot be read by anyone, thus adding a layer of security for intruder to decrypt the message End Step 4. End

3.6 PSO algorithm

This algorithm is used to select the best pixel for hiding the secret data. The optimal position is found through the PSO algorithm [23], which performs initialization by randomly selecting particles and then looks for the best fit by updating all particles' positions. Firstly, the population is selected randomly, and the particles' position is updated using an objective function. Finally, the best fit gets identified for embedding data.

3.7 User interface design

'Embedding HPSO' button on the designed interface is used for embedding the secret message in the original cover image. The text message, embedding key, and the cover image after decomposition to 2 levels using DWT serve as an input to the proposed model. The stego-image becomes the input to the 'Apply Extraction'

button, and then Huffman decoding takes place; after that, the pre-shared key at the receiver end is verified. If there is an exact match to the key, then the original message is extracted; otherwise, an encrypted image is extracted (refer Fig. 2).

4 Results and discussions

4.1 Cover images

Here, the software used is MATLAB. Four RGB images such as "peppers.jpg", "house.jpg", "mandrill.jpg", "splash.jpg", are used in this work. The original cover images used are shown in the table below (Table 1).

4.2 Performance measurements

The performance analysis can be made using the following four parameters: PSNR, MSE, SSIM and BER, which are used to appraise image quality.

1. Peak Signal to Noise ratio: PSNR is mathematically expressed as in Eq. (8). Higher the ratio, the better the quality of the image.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (8)$$

2. Mean Square Error: MSE is the other way round of PSNR. It is expressed mathematically as in Eq. (9). Lower the values better the image quality.

$$MSE = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n ||O(i,j) - s(i,j)||^2 \quad (9)$$

3. Structural Similarity Index: SSIM measures the similarity between two images, i.e., stego image and cover image. It is defined by Eq. (10) as follows:

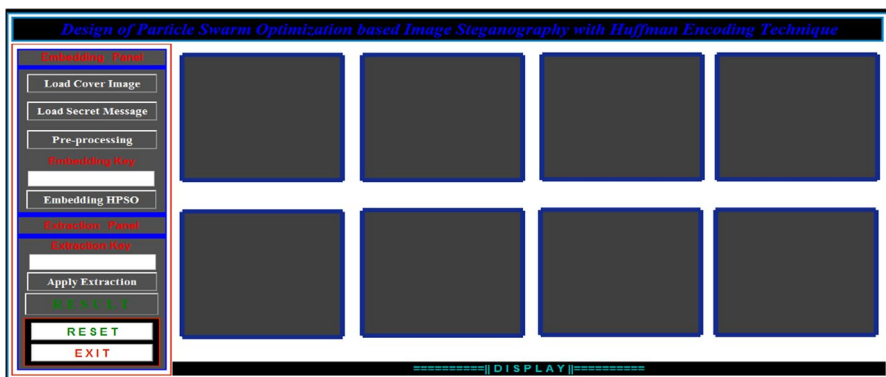


Fig. 2 Embedding and Extraction Process Screen

Table 1 Original images

Image	Image name	Image Size	Resolution
	Peppers.jpg	54.5 KB	512 × 512
	House.jpg	61.6 KB	512 × 512
	Mandrill.jpg	97.4 KB	512 × 512
	Splash.jpg	41.9 KB	512 × 512

$$SSIM = \frac{(2\mu_x\mu_y + c1)(2\sigma_{xy} + c2)}{(\mu_x^2 + \mu_y^2 + c1)(\sigma_x^2 + \sigma_y^2 + c2)} \quad (10)$$

where μ_x, σ_x^2 is the average and variance x respectively, μ_y, σ_y^2 is the average and variance of y respectively, σ_{xy} is the covariance of x and y, c1, and c2 are constants with values $c1 = m_1L^2$ and $c2 = m_2L^2$ with m_1 and m_2 having values 0.001 and 0.003 respectively and L being the range of pixels.

4. Bit Error Rate: BER is the rate at which the error occurs while transmitting the data digitally. It is expressed as in Eq. (11):

$$BER = \frac{N_{Err}}{N_{bits}} \quad (11)$$

where, N_{Err} is the no of bits received error and N_{bits} is the total no of bits. It is calculated after the complete simulation of the proposed model and is used to validate the proposed system that is truly extracted during the extraction with respect to the secret data's embedding bits pattern.

The results being considered are used with a variable set of images of multi sizes, variable message size with the application of various performance measurement metrics such as PSNR, MSE, SSIM, and BER. The proposed algorithm was tested

with different inputs to ensure that they work correctly (Refer to Table 1). The visual quality of the image is as critical as the analysis of other factors of quality.

4.3 Comparison

Here, the comparison of the proposed Huffman–Particle Swarm Optimization algorithm (HPSO) is done with the particle swarm optimization algorithm (PSO). Table 3 shows the performance metrics values obtained by using both the algorithms. It can be inferred that the proposed algorithm has a better PSNR value of 78.65 dB, where particle swarm optimization has just 41.71. The mean square error values of the proposed algorithm are less, i.e., 0.001, and it presents a less error rate of 0.035. The value of the proposed algorithm's structural similarity index is 0.999, which is almost approaching 1. This means there are very slight changes in the original image and steganographic image.

Table 4 refers to the comparison of the proposed algorithm with the previous works. First of all, comparing the proposed algorithm with reference numbers [14], PSNR values of the proposed algorithm are better, i.e., 78.65 dB, whereas authors in [6] achieved only 47.6 dB. The next parameter under consideration is MSE. The values of [14] are 0.75, where the proposed algorithm has achieved 0.00019. Now, the second comparison of the proposed algorithm is with reference number [13]. The values of PSNR and MSE for [13] are 75.12 and 0.00191, respectively, where the proposed algorithm has achieved more PSNR value, i.e., 78.65 and lower MSE value, which is 0.00019.

From the above work, it can be inferred that the Huffman PSO algorithm has the edge over other techniques and is better for the following reasons:

1. Table 2 shows the analysis of HPSO with different images and the different sizes of embedded secret messages.
2. Table 3 shows that HPSO has better quality metrics than PSO.
3. Table 4 compares the proposed steganography tool with earlier proposed techniques for image optimization.
4. Fig. 3 visually compares steganographic images with Table 1, which includes the original cover image.

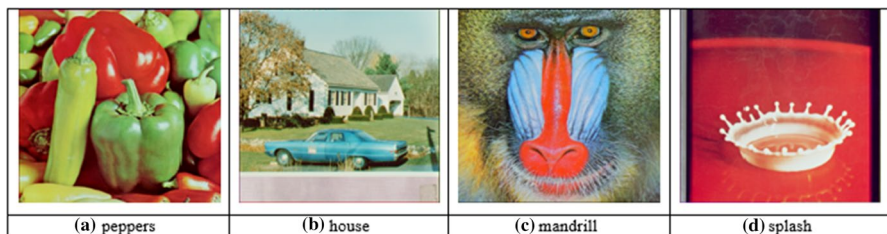


Fig. 3 Steganographic images

Table 2 Performance analysis of the proposed method

Image name	Message size (kb)	PSNR	MSE	BER	SSIM
Peppers	50	80.88	0.00017	0.045	0.987
	100	80.02	0.00015	0.051	0.983
	150	78.23	0.00018	0.043	0.991
	200	78.12	0.00019	0.035	0.999
House	50	81.95	0.00018	0.047	0.982
	100	80.14	0.00019	0.053	0.986
	150	79.23	0.00018	0.041	0.994
	200	78.66	0.00015	0.038	0.995
Mandrill	50	80.32	0.00019	0.054	0.995
	100	80.93	0.00015	0.043	0.984
	150	78.23	0.00017	0.049	0.986
	200	79.33	0.00018	0.040	0.992
Splash	50	82.11	0.00017	0.051	0.981
	100	80.75	0.00015	0.048	0.979
	150	78.88	0.00018	0.053	0.984
	200	78.45	0.00019	0.042	0.983

Table 3 Comparison of proposed algo with existing techniques

Parameters	PSO	Proposed Algo
PSNR	41.7101	78.65
MSE	4.420	0.001
BER	0.047	0.035
SSIM	0.979	0.999

Table 4 Comparison of the proposed technique with earlier previous work

Method	Year	Algorithm	PSNR	MSE
Nipankar[15]	2017	PSO & DWT	47.6	0.75
Ziyad [16]	2017	HPSO	75.12	0.00191
Proposed HPSO		Huffman PSO	78.65	0.00019

4.4 Proposed algorithm testing

The results are considered using different images (colored) with variable image sizes, variable image types, variable message length-size with an application of performance parameters such as PSNR, MSE, SSIM, and BER. The process's core purpose has been tested with several variable inputs to ensure better effectiveness and correctness (refer to Table 2). The visual quality of the image is as critical as the analysis of other factors of quality. Table 1 shows the original cover images, and Fig. 1 represents the steganographic images after the embedding process. The



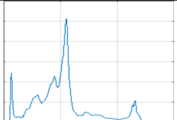
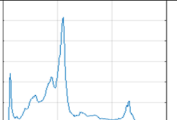


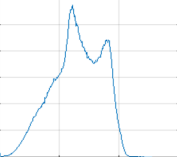
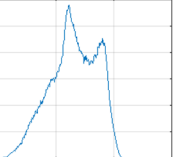


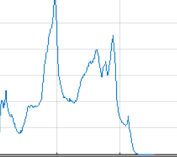





histograms of both the original and steganographic images are compared (refer to Table 5). The proposed algorithm is being run on 100 images, where the average value of PSNR, MSE, BER, and SSIM are 77.94, 0.00017, 0.045, and 0.987, respectively. The images used in this work are taken from SIPI [24] database. This database contains a collection of digital images maintained to support research in image processing, image analysis, and machine vision.

5 Conclusion

Steganography is executed using discrete wavelet transform, PSO, and Huffman encoding. The original (cover) image is disintegrated into two levels, and Huffman compression is applied on each block for high embedding capacity based on the Huffman dictionary. PSO (Particle swarm optimization) technique is used to find the best possible location in the image to embed the secret data and for further optimization purposes. The proposed algorithm is compared to existing techniques and previous research work. The results have shown that the proposed algorithm performs better in terms of PSNR, MSE, BER, and SSIM. On various parameters and conditions, the proposed algorithm has achieved the average values of 77.94, 0.00017, 0.045, and 0.987 for the parameters like PSNR, MSE, BER, and SSIM, respectively.

This research focuses on addressing image steganography's major concerns, i.e., data embedding capacity, robustness, and imperceptibility. This paper does not

Table 5 Histogram Analysis of both original image and steganographic image

Original image	Steganographic image	Histogram of original image	Histogram of steganographic image
			
			
			
			

address the exchange of a secret key between sender and receiver since it is pre-shared between them. Therefore, in the future, we would like to work on the key exchange mechanisms. In order to solve the problem of high payload, other optimization algorithms can also be considered.

6 References

1. FA.HD. Mohsen, M. Hadhoud, K. Mostafa, K. Amin, (2012) A new image segmentation method based on particle swarm optimization. *Int. Arab J Inform Technol*, **9**(5)
2. A. Aggarwal, S.K. Patra, Performance prediction of OFDM based digital audio broadcasting system using channel protection mechanisms. *Int. Conf. Electron. Comp. Technol* **2**, 57–61 (2011)
3. S.K. Sabnis, R.N. Awale, Statistical steganalysis of high capacity image steganography with cryptography. *Proc. Comput. Sci.* **79**, 321–327 (2016)
4. M. Jain, S.K. Lenkab, S.K. Vasisthaa, Adaptive circular queue image steganography with RSA cryptosystem. *Perspect. Sci.* **8**, 417–420 (2016)
5. S.U. Mahaeshwari, D.J. Hemanth, Performance enhanced image steganography systems using transforms and optimization techniques. *Multimedia Tools Application* (2015). <https://doi.org/10.1007/s11042-015-3035-1>
6. W. Hong, T.S. Chen, Reversible data embedding for high-quality images using interpolation and reference pixel distribution mechanism. *J. Vis. Commun. Image Represent.* **22**, 131–140 (2011)
7. A. Sharif, M. Mollaeefar, M. Nazari, A novel method for digital image steganography based on a new three-dimensional chaotic map. *Multimed. Tools Appl.* **76**(6), 7849–7867 (2017). <https://doi.org/10.1007/s11042-016-3398-y>
8. M.S. Subehdar, V.H. Mankar, Image steganography using redundant discrete wavelet transform and QR factorization. *Comput. Electr. Eng.* **54**, 406–422 (2016)
9. S. Yan, G. Tang, Y. Chen, Incorporating data hiding into G. 729 speech codec. *Multimed. Tools Appl.* **75**(18), 493–512 (2016)
10. G. Swain, Digital image steganography using variable length group of bits substitution. *Proc Comput Sci* **85**, 31–38 (2016)
11. U. Dewangan, M. Sharma, S. Bera, Development and analysis of stego image using discrete wavelet transform. *Int. J. Sci. Res.* **2**(1), 142–148 (2013)
12. A. Pradhan, A. K. Sahu, G. Swain, K. R. Sekhar, Performance evaluation parameters of image steganography techniques, *Proceedings of International Conference on Research Advances in Integrated Navigation Systems*, 1–8, 2016
13. Z.T.M. Al-Ta'i, E.R. Mohammad, Comparison between PSO and HPSO in Image Steganography. *Int. J. Comput. Sci. Inf. Secur.* **15**(8), 161–168 (2017)
14. S.I. Nipanikar, V. Hima Deepthi, N. Kulkarni, A sparse representation based image steganography using Particle Swarm Optimization and Wavelet transform. *Alex. Eng. J.* (2017). <https://doi.org/10.1016/j.aej.2017.09.005>
15. A. Miri, K. Faez, Adaptive image steganography based on transform domain via genetic algorithm. *Optik-Int. J. Light Electron Optics* (2017). <https://doi.org/10.1016/j.ijleo.2017.07.043>
16. P. Rajeswari, P. Shwetha, S. Purushothaman (2017, March) Application of wavelet and particle swarm optimization in steganography. In 2017 2nd International Conference on Anti-Cyber Crimes (ICACC) (pp. 129–132). <https://doi.org/10.1109/Anti-Cybercrime.2017.7905277>
17. I. Hamid, Image steganography based on discrete wavelet transform and chaotic map. *Int. J. Sci. Res.* (2018). <https://doi.org/10.21275/ART20179396>
18. M. Khari, A.K. Garg, R.G. Crespo, E. Verdú, Gesture recognition of RGB and RGB-D static images using convolutional neural networks. *Int. J. Interact. Multimed. Artif. Intell.* **5**(7), 22 (2019)
19. R. Gupta, M. Khari, D. Gupta, R.G. Crespo, Fingerprint image enhancement and reconstruction using the orientation and phase reconstruction. *Inf. Sci.* (2020). <https://doi.org/10.1016/j.ins.2020.01.031>
20. R. Gupta, M. Khari, V. Gupta, E. Verdú, X. Wu, E. Herrera-Viedma, R. González Crespo, Fast single image haze removal method for inhomogeneous environment using variable scattering coefficient. *Comput. Modeling Eng. Sci.* **123**(3), 1175–1192 (2020)

21. M. Khari, A. Sinha, E. Verdu, R.G. Crespo, Performance analysis of six meta-heuristic algorithms over automated test suite generation for path coverage-based optimization. *Soft Comput.* (2019). <https://doi.org/10.1007/s00500-019-04444-y>
22. M. Khari, P. Kumar, D. Burgos, R.G. Crespo, Optimized test suites for automated testing using different optimization techniques. *Soft Comput.* **22**(24), 8341–8352 (2018)
23. F. Mohsen, M.M. Hadhoud, K. Moustafa, K. Ameen, A new image segmentation method based on particle swarm optimization. *Int. Arab J. Inform. Technol.* **9**(5), 487–493 (2012)
24. <http://sipi.usc.edu/database/>-The USC-SIPI Image Database

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.