Multimedia Tools and Applications (2021) 80:18821–18837
https://doi.org/10.1007/s11042-021-10608-6

Towards a high capacity coverless information hiding approach

Fatimah Shamsulddin Abdulsattar[1]

Received: 1 March 2020 / Revised: 7 October 2020 / Accepted: 25 January 2021 /
Published online: 20 February 2021

Abstract
Most of the coverless information hiding approaches use a set of cover images as stego-images which make them unsuitable for real-time applications. Moreover, the parameters that affect the hiding capacity and the robustness against image processing attacks are not explicitly studied. This paper explores the effectiveness of coverless information hiding using only one cover image to transmit secret information based on eigen decomposition. The proposed approach performs coverless information hiding by establishing mapping relationships between the hash codes of the image blocks and the characters of the secret message. The hash code is calculated by splitting the block into 9 sub-blocks and then comparing the largest eigenvalues of the sub-blocks according to four arrangements. To speed up the embedding process, we build a lookup table to store the pre-computed hash codes with the corresponding block locations. The approach has three important parameters: overlapping blocks, arrangements of sub-blocks, and block sizes. Several experiments are conducted to analyze the effect of these parameters on performance. The results of the analyses indicate that the overlapping between image blocks is necessary to generate a sufficient number of unique hash codes. Otherwise, the embedding process could not be completed using a single image. The arrangement between sub-blocks has a lower impact on the number of unique hash codes and the resilience against image processing attacks. The block size is another important parameter. As the block size increases, the hiding capacity decreases and the resilience against image processing attacks improves. Compared with other coverless information hiding approaches, the proposed approach has a higher hiding capacity and better resilience against image processing attacks. Moreover, the approach has short execution time and better resistance to detecting tools.

Keywords Eigen decomposition · Information hiding · ASCII-codes · Hiding capacity

1 Introduction

Digital multimedia data is transmitted over public channels with the rapid advancement of digital technology and computer networks [20]. Considerable efforts have thus been spent

✉ Fatimah Shamsulddin Abdulsattar
fsa@uomustansiriyah.edu.iq

[1] Computer Engineering Department, Engineering College, Mustansiriyah University, Baghdad, Iraq

for protecting the transmission over public channels. Cryptography has been employed for securing multimedia data but after decryption, the security will be violated. Information hiding is therefore suggested to tackle the deficiencies [18, 24]. Information hiding is a technique which conceals data into digital media like image, text, voice and video, this technique has become a significant topic in this field [7]. Since image represents one of the important carriers used in multimedia transmission, several works have been done in image steganography. In general, the traditional information hiding techniques conceal information by slightly changing the carrier. These techniques are either implemented in the spatial domain [6, 13, 15] or in the spectral domain [5, 8, 19]. Steganalysis, as an opponent to information hiding, attempts to discover the existence of secret information in a mistrust carrier using signs of alteration remained in the carrier. The steganalysis (detection) methods use statistical characteristics for detection. Although the secret information concealed in the carrier is difficult to notice by human eyes, the detection methods can use the traces of modifications to discover the presence of secret information [18].

Accordingly, the coverless information techniques were introduced to withstand steganalysis methods. Unlike the conventional information hiding techniques, the coverless information hiding idea is to select cover images with features representing secret information as stego-images [2]. Mapping relationships can possibly be built between image features and secret message segments using a powerful hashing algorithm [3, 23]. The hash algorithm is used for generating one or more hash sequences from the image by dividing it into several blocks. In general, the existing coverless information hiding approaches have three limitations. First, the information hiding techniques that are based on mapping relationships use multiple stego-images in order to transmit one secret message. As the message length gets longer, the number of stego-images increases accordingly, making such techniques impractical. Second, none of the studies explores the effect of the cover image block size and the overlap between blocks on the resilience against image processing attacks. Third, the relationship between the arrangement of the neighboring sub-blocks when extracting the hash sequence and the performance of the technique has not been studied.

To tackle the above issues, this paper develops a new coverless information hiding approach using a single cover image. The hash sequences of the cover image are calculated using eigen decomposition to improve their immunity against image processing attacks since this tool can be employed for estimating covariance between image pixels. This paper has the following contributions:

(1) Instead of building a database prior to sending a single secret message via a coverless information hiding approach, we explore the efficiency of coverless information hiding strategy using a single image for transmitting a complete secret message. (2) The efficiency of using eigen decomposition to calculate the hash codes from the cover image blocks have been studied. (3) Extensive experiments that analyze the effect of three parameters (block sizes, arrangements of neighboring sub-blocks, and overlapping strategy) on coverless information hiding performance have been conducted. The remaining of the paper is managed as follows: Related studies are discussed in Section 2. Preliminaries are given in Section 3. The proposed coverless information hiding approach is given in Section 4. Experimental results and analyses are given in Section 5. The conclusions are made in Section 6.

2 Related works

Images are popular media to hide data in coverless information hiding approaches using mapping rules. The efficiency of these approaches is based on the power of the adopted

hashing algorithm. Several approaches are developed in this field. Zhou et al. [20] introduced an image steganography technique by matching the grey values in the image with the secret data. In this technique, a database is established and a set of images which match the secret data is then chosen as stego-images. Each stego-image can conceal 8 bits of secret data. Zhou et al. [21] developed a new information hiding approach using the Bag of Words (BOW) model. In this approach, a set of visual words that represent the secret data is extracted from this model and transmitted to the receiver. A new hashing algorithm based on the histogram of oriented gradients is developed in [22], which is used for image steganography. The technique chooses a number of images, whose hash sequences match the secret information pieces from a constructed database. These images are used for secret communication. To improve the hiding capacity of the method [20], an efficient image hashing algorithm is introduced in [17] using Scale-Invariant Feature Transform (SIFT). The hash sequence of each carrier image in this algorithm can represent 18 bits of secret information.

After that, Yan et al. [14] combined SIFT and Bag-of-Features (BOF) to generate image hash sequences in order to realize coverless information hiding. To improve the resistance capability against image processing attacks, Zhang et al. [16] developed a modern image steganography method using Discrete Cosine Transform (DCT) and Latent Dirichlet Allocation (LDA). This method used LDA for classifying an image database according to their topics. After that, the hash sequence is calculated from each image in the DCT domain. This method can hide between 1-15 bits of information in a single image. In [25], a recent image steganography technique using mapping relationships is introduced. This technique calculates hash sequences by dividing each image into 9 blocks. Then, the mean value of each block is calculated and compared with the mean values of the adjacent blocks to produce the final hash sequence of the image. This technique is dedicated to transmitting Chinese sentences. A recent coverless steganography method is developed in [24] for transferring a secret image. The method works by sending a number of carrier images which have similar blocks to the secret image from a previously built database.

Instead of extracting low-level features, deep learning is recently employed for calculating high-level features from carrier images. Deep generative adversarial networks are used by Hu et al. [4] for developing a modern image steganography method. In this method, the secret information is matched with a noise vector. Then, the carrier image is generated by training a generative neural network according to the noise vector. Faster Region-based Convolutional Neural Networks (Faster-RCNN) is employed in [18] for designing an efficient information hiding approach. The Faster-RCCN is utilized for determining the locations of objects in images. After that, the secret information can be represented using the labels of these objects. Convolutional networks-based information hiding approach is also proposed in [7] for transferring secret images. This approach used CNN to extract semantic features from image blocks in a constructed database. In order to transfer a secret image, a series of cover images whose blocks have a visual similarity to the secret image is sent to the receiver.

3 Preliminaries

3.1 Eigenvalues and eigenvectors

Eigenvalues and eigenvectors are often used to calculate the orthogonal linear transformation of square matrices. They have their importance in computer vision applications such as image classification and object recognition. In an image, eigenvalues and eigenvectors are used to estimate the covariance between its pixels. The largest eigenvalue of the image

is related to the dominant eigenvector, which depicts the direction of the maximum variation between image pixels. To find the eigenvectors of a matrix B, the following condition should be held [1]

$$Bw_i = \gamma_i w_i \tag{1}$$

Where γ_i is an eigenvalue and w_i is the corresponding eigenvector. The condition for solving the above equation is

$$|B - \gamma_i I| = 0 \tag{2}$$

Where I is an identity matrix with the same size as B. In the proposed approach, eigenvalues are calculated from each image block and the largest eigenvalue is then determined for calculating the hash codes.

3.2 Paillier public key encryption

Paillier cryptosystem uses public-key cryptography and is invented by Pascal Paillier in 1999. It is commonly employed in the areas of secure computation. Using this system, semantic security and multiplicative homomorphism property can be achieved [10]. To generate the public and private key, we choose two large prime numbers p and q that satisfy the condition $gcd(p \times q, (p-1) \times (q-1)) = 1$ and then calculate $n = p \times q$ and $\lambda = lcm(p-1, q-1)$ where $lcm(.)$ is the least common multiplier. After that, we choose another random integer $g \in Z^*_{n^2}$ which satisfy the condition $gcd(L(g^\lambda \bmod n^2), n) = 1$, where $L(x) = (x-1)/n$, $Z_{n^2} = \{0, 1, 2...n^2\}$ and $Z^*_{n^2}$ contains all elements in Z_{n^2} which are relatively prime with n^2. Finally, we use (g, n) as a public key and (λ) as a private key. For encryption, suppose p is a plaintext to be encrypted where $p \in Z_n$. Then, the sender calculates its corresponding ciphertext c as

$$c = g^p r^n \bmod n^2 \tag{3}$$

On the other hand, the plaintext can be retrieved from the ciphertext by

$$p = (L(c^\lambda \bmod n^2))/(L(g^\lambda \bmod n^2)) \bmod n \tag{4}$$

In the proposed approach, this algorithm will be used to encrypt the common information between the sender and receiver. The locations of the image blocks, which will be employed for embedding the secret message are stored in a separate file. The location of each image block consists of two pieces: x-coordinate and y-coordinate. The location information file is encrypted using the Paillier encryption algorithm before transmitting to the receiver. The size of the location file will be (the length of the secret message characters $\times$ 2). Another way to store the location information is by numbering the cover image blocks as 0, 1, 2, ... etc. In this case, the size of the location file is the same as the length of the secret message characters. To provide another layer of security, the location file can be permutated by any means of permutation methods and sent over a secure channel.

4 The proposed information hiding approach

The proposed approach consists of four main parts: hash code generation, lookup table establishment, information embedding and information extraction. The principal idea of the proposed approach is based on building mapping relationship between hash codes extracted from cover images and secret message characters. The hash codes are calculated from image

blocks using eigen decomposition. Unlike other coverless information hiding approaches, only one stego-image is used for transmitting a secret message in this work. In order to do this, several unique hash codes should be extracted from a cover image. Therefore, the proposed approach splits the cover image into partly overlapping blocks. Although the resilience towards image processing attacks could be affected when using overlapping blocks for hiding secret information, several advantages can be achieved: (1) the hiding capacity will be increased, (2) the bandwidth required to transmit the secret message will be reduced, (3) the extensive image searching and indexing operations will be eliminated, (4) no large image database is required.

From each block in the cover image, fixed-length binary hash code is extracted and converted to its corresponding ASCII-code. Each character of the secret message is also converted to its ASCII-code. The embedding is achieved by matching each character in the secret message with the extracted hash code from the image block based on their ASCII-code. The location information of the matching blocks is placed in a file. After matching all the characters of the secret information with the image blocks, the image is sent to the receiver as a stego-image and the location information file is encrypted and used as a shared secret key between the two communicating parties (i.e. sender and receiver). When the stego-image is transmitted, the receiver decrypts the location information file to determine the target image blocks and calculate the binary hash codes. Finally, the hash codes are converted to their corresponding ASCII-codes to retrieve the characters of the secret message. Figure 1 illustrates the steps of the proposed approach.

4.1 Calculation of a binary hash code

Hash code is a fixed-length binary sequence, which is calculated from an image block based on its local content. Each image block can be expressed by a hash code, which can then be related to a specific part in the secret information. In this work, the hash code is calculated using eigen decomposition This decomposition has several applications in machine learning and computer vision fields such as object recognition and dimensionality reduction. The covariance between image pixels can be depicted using eigen decomposition. The maximum variation between image pixels can be described using the largest eigenvalue and its

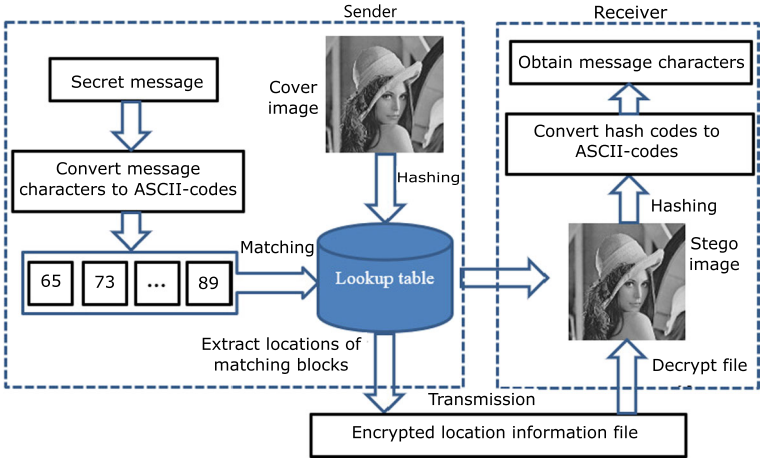

Fig. 1 The layout of the proposed information hiding approach

corresponding eigenvector. The steps to calculate the hash code from each image block are presented in Fig. 2 and the following steps give more details.

Step 1: The cover image is split into blocks with size $L \times L$ as

$$S_1, S_2 ... S_T \tag{5}$$

where T is the number of blocks.

Step 2: Each block S_i is further split into 9 sub-blocks with size $H \times H$. The size of the sub-block ss_j is determined as

$$\text{size}(ss_j) = H \times H = \frac{L}{3} \times \frac{L}{3} \tag{6}$$

Step 3: The largest eigenvalue e_{maxj} is calculated from each sub-block ss_j as

$$e_{maxj} = \max\{e_k\}, \quad k = 1, 2 ... z \tag{7}$$

where $max\{.\}$ is the maximum function and z is the number of eigenvalues of a sub-block.

Step 4: The hash code c_{si} for each block S_i is calculated by comparing the largest eigenvalues e_{maxj} of the adjacent sub-blocks according to one of four arrangements. The order of the sub-blocks in each of these arrangements are illustrated in Fig. 3.

The hash code is calculated depending on the arrangements 1, 2 and 3 [see Fig. 3] by comparing the largest eigenvalues of the adjacent sub-blocks as

$$c_{1_i} = \begin{cases} 1 & \text{if} \quad e_{maxj} \geq e_{maxj+1} \\ 0 & \text{Otherwise.} \end{cases} \qquad 1 \leq j \leq 8 \tag{8}$$

On the other hand, the hash code is calculated depending on the arrangement 4 [see Fig. 3] by comparing the largest eigenvalue of the central sub-block with the largest eigenvalues of the surrounding sub-blocks as

$$c_{2_i} = \begin{cases} 1 & \text{if} \quad e_{max5} \geq e_{maxj+1} \\ 0 & \text{Otherwise.} \end{cases} \qquad 1 \leq j \leq 8 \quad \text{and} \quad j \neq 5 \tag{9}$$

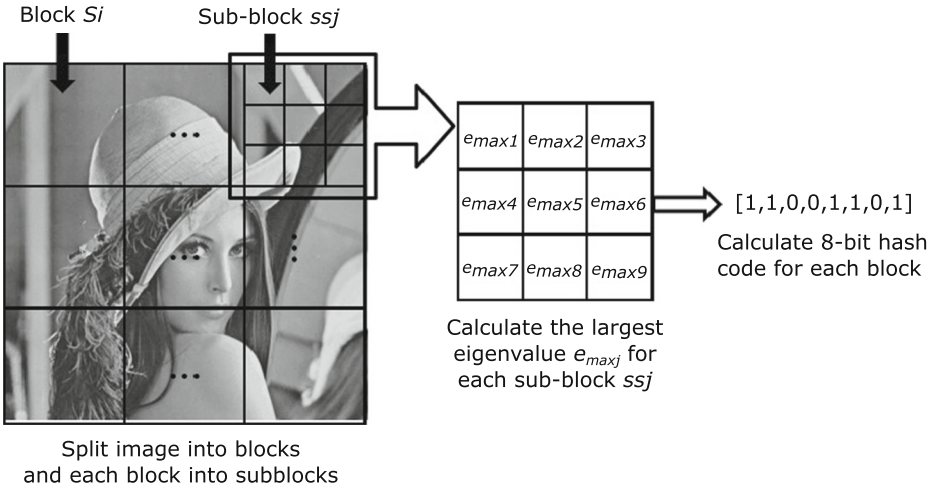

Fig. 2 The process of calculating hash code

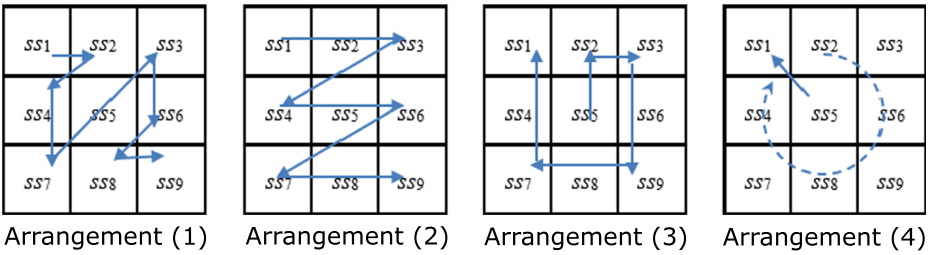

Fig. 3 The arrangement of neighboring sub-blocks

4.2 Establishment of a lookup table

Several 8-bit hash codes can be generated from the image. The number of these codes is the same as the number of image blocks. These hash codes are converted to their ASCII-codes to be used in the embedding process. During information embedding process, each character in the secret message is converted to its ASCII-code value and can then be mapped to one or several image blocks, which have the same hash code as the message character. To speed up the search of a particular image block that corresponds to each message character, a lookup table is established for the cover image. The lookup table is used for storing the location of each image block and its corresponding hash code after converting it to its ASCII-code. The location of the block in the image is determined from the coordinate of the first (upper left corner) pixel in the block. There is also an additional field, which will be employed as a mask to indicate whether or not the current block has been employed for embedding data. Let the coordinate of the last block in the cover image is (p, p), Fig. 4 demonstrates the layout of the lookup table.

4.3 Information embedding

The algorithm of information embedding is implemented at the sender side to hide secret information in a suitable cover image, which will be sent to the receiver as a stego-image. To improves the security of the approach, the sender should select a new cover image for each new secret message. The embedding procedure involves the following steps:

Step1: The cover image is resized to a predetermined size $(M \times N)$ in order to make the approach resistance to rescaling attack.

Step2: The sender calculates the hash codes from the image blocks and establishes a lookup table to store the calculated hash codes (in ASCII-value) with their associated block positions to speed-up the embedding process.

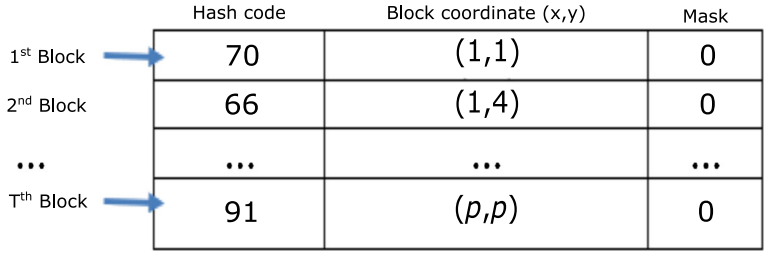

Fig. 4 The lookup table structure

Step3: During embedding process, each character in the secret message is first converted to the corresponding ASCII-code and then matched with the appropriate image block(s), which has the same hash code in the lookup table and the location information of the matching block is placed in a file. The mask field of the matching block is set to one in the lookup table. If the message character matches multiple blocks, then the approach selects the image block that is not previously employed using the mask field to improve the security of the embedding. Thus, if the same character appears multiple times in the message, then it can be matched with a different image block each time.

Step4: When all message characters are matched with their corresponding image blocks, the location information file is encrypted using the Paillier public-key encryption algorithm. The encrypted file represents a shared secret key between the two communicating parties (sender and receiver) and the cover image is sent to the receiver side as a stego-image.

4.4 Information extraction

To retrieve the hided message characters from the stego-image, the receiver does the following information extraction procedure

Step1: The stego-image is first resized to the pre-determined size ($M \times N$).

Step2: The location information file is decrypted to get the positions of the target image blocks in the stego-image.

Step3: The receiver split each target block into 9 sub-blocks to calculate the hash codes.

Step4: The largest eigenvalue e_{maxj} of each sub-block is calculated and the hash code of the block is calculated using either (8) or (9) according to which arrangement is considered.

Step5: The binary hash code of the block is converted to its corresponding ASCII-code to obtain the character of the secret message. Then, **Step3**-**Step5** are repeated to obtain the remaining characters.

5 Results and discussion

Several experiments are accomplished to explore the effectiveness of our information hiding approach using different neighborhood arrangements and a various number of image block sizes. In all experiments, the image blocks are partially overlapped by two-third block width from the upper left corner to the lower right corner, which is found to be sufficient to produce as many unique hash codes as required for embedding one message in a single image. The size of the embedding message in all experiments is 1.16 KB unless it states otherwise. We used six grayscale images of size 512×512. These images are Lenna, Baboon, Cameraman, Goldhill, Houses and Boats, as shown in Fig. 5. All the experiments are implemented on a computer with the following specifications: Intel(R) Core™ i7-4800MQ CPU @ 2.70 GHz, 16GB memory, and Windows 10 (64 bits). All the codes are written in MATLAB 2019a. The analysis concentrates on the number of unique hash codes, execution time, resilience against image processing attacks and information hiding capacity.

5.1 Analysis the number of unique hash codes

The hash codes calculated from a cover image is an important factor in the embedding process. When the calculated hash codes are varied and cover a broad range of different values, the probability of embedding a secret message with different characters in a single image increases. Since the length of a hash code calculated from each block is 8 bits. Therefore,

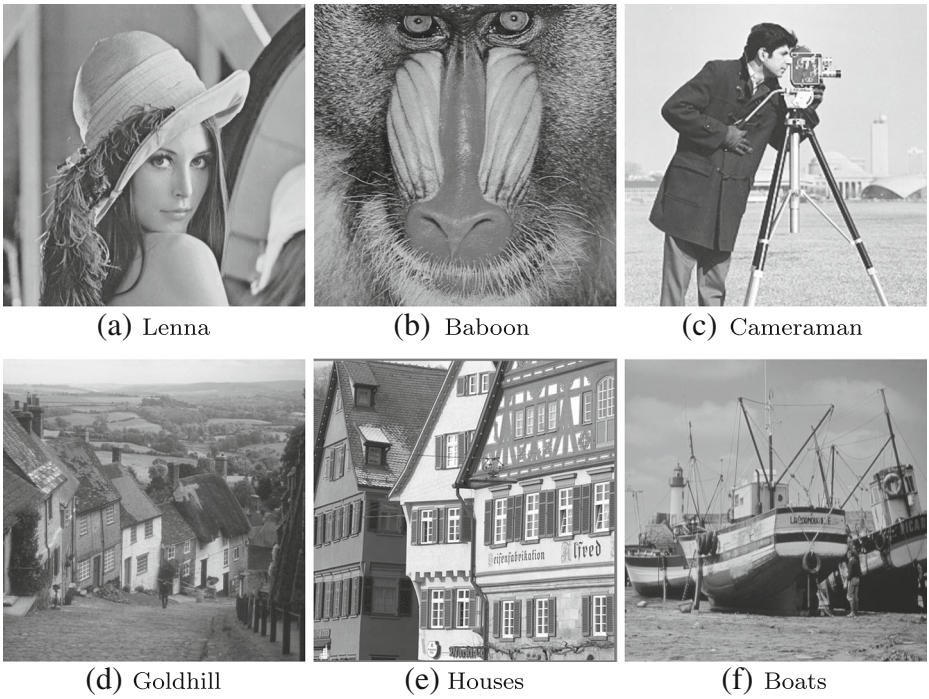

Fig. 5 The test cover images

the maximum number of different (unique) hash codes would be $2^8 = 256$. Several experiments that analyze the number of unique hash codes with respect to various parameters are conducted. In the beginning, the relationship between a sub-block and block size is studied. Table 1 displays an example of several sub-block sizes in the range between 3×3 and 8×8 and their corresponding block sizes. As can be observed, the block size is three times the sub-block size.

(A) The block size versus unique hash codes The first set of experiments are established to explore the relationship between the number of unique hash codes and the block size using all the six images. To this end, each image is split into partially overlapping blocks. Each block is further split into 9 sub-blocks. The size of the block in each experiment is chosen as indicated in Table 1. We choose the second arrangement (arrangement (2) in Fig. 3) when calculating the hash code from each block. Table 2 summarizes the results of these experiments. The table indicates that the number of unique hash codes decreases as the block size increases for most cover images. When the block size is larger than 18×18, the number of unique hash codes calculated from each image varies, which is probably due to the various details of the images. When the image contains many different details (e.g.

Table 1 The relationship between sub-block and block size

Sub-block size	3 × 3	4 × 4	5 × 5	6 × 6	7 × 7	8 × 8
Block size	9 × 9	12 × 12	15 × 15	18 × 18	21 × 21	24 × 24

Table 2 The number of unique hash codes versus block size

Image	Block size					
	9 × 9	12 × 12	15 × 15	18 × 18	21 × 21	24 × 24
Lenna	255	255	250	231	220	214
Baboon	256	256	255	252	245	241
Cameraman	256	255	256	255	251	249
Goldhill	256	256	256	255	251	251
Houses	256	256	256	256	252	253
Boats	256	256	256	256	254	254

Goldhill, Boats and Houses), the number of unique hash codes is higher than that obtained from images with uniform details. In general, the block size in the range between 3×3 and 18×18 can produce a sufficient number of unique hash codes for most images.

(B) The arrangement of neighboring sub-blocks versus unique hash codes This set of experiments study the effect of the arrangement of neighboring sub-blocks on the number of unique hash codes calculated from each cover image. Table 3 depicts the relationship between the arrangement of neighboring sub-blocks and the number of unique hash codes for three different images (Lenna, Boats, Baboon). As can be observed, there are differences in the number of unique hash codes calculated over different arrangements. These differences are not significant for the first three block sizes and they become significant when the

Table 3 The number of unique hash codes versus the arrangements of sub-blocks

Image	Block size	Arrang. (1)	Arrang. (2)	Arrang.(3)	Arrang.(4)
Lenna	9 × 9	**256**	255	255	256
	12 × 12	**255**	255	251	254
	15 × 15	**253**	250	244	250
	18 × 18	**247**	231	244	241
	21 × 21	237	220	**243**	240
	24 × 24	231	214	**235**	226
Boats	9 × 9	254	**256**	254	256
	12 × 12	250	**256**	252	256
	15 × 15	245	**256**	248	246
	18 × 18	235	**256**	248	240
	21 × 21	228	**254**	244	236
	24 × 24	220	**254**	240	231
Baboon	9 × 9	253	256	252	**256**
	12 × 12	255	256	252	**256**
	15 × 15	252	255	250	**256**
	18 × 18	245	252	246	**256**
	21 × 21	247	**254**	246	252
	24 × 24	230	**254**	241	243

block size is larger than 18×18. For Lenna image, arrangement (1) gives the highest number of hash codes using the first four block sizes. Arrangement (2) produces the highest number of hash codes for Boats image. Finally, arrangement (4) is the best for Baboon image using the first four block sizes. The differences in image details could be the main reason for the variation in the results. In general, arrangement (2) produces a sufficient number of unique hash codes (≥ 250) using the first three block sizes for the three images.

(C) The overlapping and non-overlapping blocks versus unique hash codes The overlapping operation is another important factor in our proposed method. The overlapping operation allows producing a sufficient number of unique hash codes that is required to embed a secret message within a single image. Therefore, the third set of experiments is implemented to highlight the differences between overlapping and non-overlapping blocks in terms of the number of unique hash codes. To this end, two images (Lenna and Houses) with different details are selected. The second arrangement is chosen to calculate the hash codes using non-overlapping blocks and overlapping blocks by two-third block width. The differences between the two strategies are represented in Fig. 6. It is evident from this figure that the number of unique hash codes produced by overlapping blocks is higher than that calculated by non-overlapping blocks for the two images. This number decreases as the block size increases and the rate of reduction varies between the two images. Consequently, the overlapping operation is necessary if the embedding process has to be carried out using single images.

5.2 Execution time analysis

This section analyzes the execution time of hashing, embedding, and extracting processes using different block sizes and message lengths. To this end, we used two secret messages with different lengths. The hash codes are calculated using overlapping blocks of sizes 9×9, 12×12, 15×15, and 18×18. Table 4 summarizes the execution time (in seconds) of each process separately. It can be observed that the execution time of the hashing process is higher than that of the embedding and extracting process because it depends on calculating the hash code of each image block. The hashing time is almost similar for both message lengths since the hashing process relies on the size of the cover image and not on the length of the secret message. On the other hand, the embedding and extracting time becomes higher when

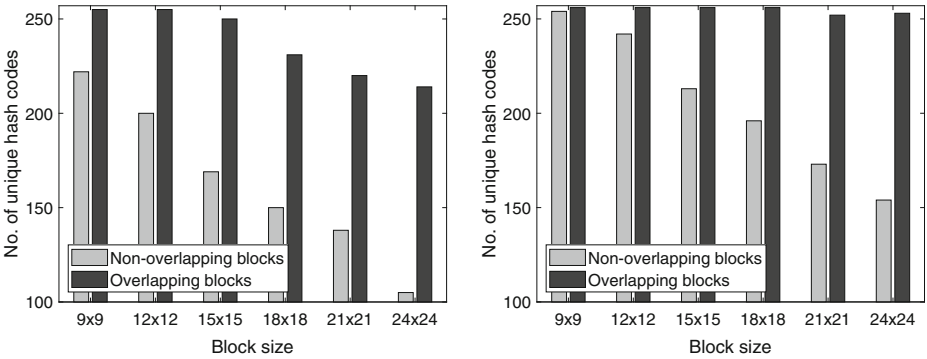

Fig. 6 The number of unique hash codes versus overlapping and non-overlapping blocks

Table 4 Execution time (seconds) of hashing, embedding and extracting process

Operation	Message length=4768 bits				Message length=9512 bits			
	9×9	12×12	15×15	18×18	9×9	12×12	15×15	18×18
Hash	6.92	3.87	2.84	1.99	6.74	4.06	2.82	2.03
Embed	**0.07**	**0.06**	**0.05**	**0.04**	**0.16**	**0.12**	**0.10**	**0.08**
Extract	0.13	0.14	0.16	0.17	0.26	0.28	0.31	0.34

the length of the message becomes bigger. The embedding process spends less time than the extracting process because the embedding process is done using the lookup table, which contains the pre-computed hash code for each image block. Additionally, the time of hashing and embedding process is inversely proportional to the block size because when the block size becomes larger, the number of blocks and their corresponding hash codes decreases and the length of the lookup table used in the embedding process becomes smaller. On the contrary, the time of extracting process becomes higher when the block size becomes larger since the computational cost for calculating the hash code from a larger block increases. In general, the total execution time of the three processes is short and our proposed hiding method is suitable for real-time applications.

5.3 Resilience against image processing attacks

During a transmission process, images could suffer from several image processing attacks which could damage some of their contents. Resilience is a vital parameter to judge the performance of information hiding techniques. It measures how properly the receiver can retrieve the secret message. The following attacks are considered in the experiments.
(a) Gaussian noise with variance $(v) = 0.001$ and 0.005.
(b) Salt-pepper noise with rate $(r) = 1\%$ and 5%.
(c) Speckle noise with variance $(v) = 0.01$ and 0.05.
(d) Mean filtering with window size (w) of 3×3 and 5×5.
(e) Median filtering with window size (w) of 3×3 and 5×5.
(f) Gaussian low-pass filtering with window size (w) of 3×3 and 5×5.
(g) Histogram equalization.

We used Bit Error Rate (BER) to assess the resilience of the approach to the attack. The BER is given as [18]

$$BER = \frac{\sum_{r=1}^{Z} \text{xor}(a_r, b_r)}{Z} \tag{10}$$

Where a_r is the hided message bit before attacks, b_r is the extracted message bit after attacks and Z is the message length in bits.

(A) The resilience in terms of the arrangement of neighboring sub-blocks The hash code is extracted from each block in the cover image by comparing the largest eigenvalue in the neighboring sub-blocks based on four arrangements. Therefore, the first set of experiments are conducted to study the relationship between the arrangement of the neighboring sub-blocks and the resilience against image processing attacks. To this end, we split the cover image into partially overlapping blocks of size 9×9. Each block is then split into 9 sub-blocks. In each experiment, the hash codes are calculated using one of the four arrangements discussed in Section 4.1. Table 5 abbreviates the results of the experiments. It can be noticed

Table 5 The resilience (%) versus arrangements of sub-blocks

Attack	Arrang.(1)	Arrang.(2)	Arrang.(3)	Arrang.(4)
Gaussian noise ($v = 0.001$)	27.09	**25.39**	28.78	27.09
Gaussian noise ($v = 0.005$)	35.69	**33.98**	37.34	35.69
Salt-pepper noise ($r = 0.001$)	**0.79**	0.91	0.82	**0.79**
Salt-pepper noise ($r = 0.005$)	**3.57**	3.69	3.65	**3.57**
Speckle noise ($v = 0.01$)	31.99	**27.85**	32.79	31.99
Speckle noise ($v = 0.05$)	40.01	**38.03**	39.18	40.01
Speckle noise ($v = 0.1$)	42.61	41.62	**40.71**	42.61
Median filtering ($w = 3 \times 3$)	16.54	**14.67**	16.67	17.24
Median filtering ($w = 5 \times 5$)	25.01	**22.48**	23.88	23.69
Mean filtering ($w = 3 \times 3$)	13.18	**13.10**	14.06	13.96
Mean filtering ($w = 5 \times 5$)	22.64	21.73	21.45	**21.12**
Gaussian filtering ($w = 3 \times 3$)	**4.01**	4.35	4.65	5.26
Gaussian filtering ($w = 5 \times 5$)	13.13	12.66	**12.64**	13.40
Histogram equalization	**4.78**	5.45	5.21	5.65

that no optimal arrangement gives superior resilience against all attacks. Arrangement (1) gives better results against salt and pepper noise, Gaussian filtering, and histogram equalization. Arrangement (2) is the optimal choice against Gaussian and speckle noise as well as median and mean filtering. Arrangement (3) shows superior resilience against speckle noise and Gaussian filtering. Finally, arrangement (4) is the best for dealing with salt and pepper noise and mean filtering. In general, arrangement (2) gives better results for a broad range of image processing attacks. Therefore, this arrangement is considered in the remaining analyses.

(B) The resilience in terms of image block sizes We explore the effect of various image block sizes on the resilience against image processing attacks. To do this, the information hiding approach is implemented using arrangement (2) to calculate the hash code from each block in the cover image. The cover image is split into partially overlapping blocks of sizes 9×9, 12×12, 15×15, and 18×18. For each size, the image block is split into 9 sub-blocks. The overlapping is necessary to generate several unique hash codes from a single image as described in Section 5.1. Table 6 describes the effect of using different image block sizes on resilience. In general, the results clarify that the resilience against most of the image processing attacks improves when the block size increases. To get a trade-off between the resilience and the number of unique hash codes, the block size of 18×18 is a better choice for our proposed approach.

(C) The resilience of different information hiding approaches The resilience of the proposed approach is compared to three other information hiding approaches in the literature: SIFT+BOF [14], DCT+LDA [16] and Faster-RCNN [18]. The previous experiments demonstrated that the proposed approach can produce a better resilience when the block size is 18×18. Therefore, the block size of 18×18 is used with the proposed approach for comparison purposes. Table 7 displays the comparison results among these approaches. This table reveals that the proposed approach has higher resilience to a wider range of image

Table 6 The resilience (%) versus block sizes

Attack	Block size			
	9 × 9	12 × 12	15 × 15	18 × 18
Gaussian noise ($v = 0.001$)	27.09	19.58	14.65	**10.39**
Gaussian noise ($v = 0.005$)	35.69	28.39	23.83	**17.77**
Salt-pepper noise ($r = 0.001$)	**0.79**	1.06	1.20	1.79
Salt-pepper noise ($r = 0.005$)	**3.57**	6.07	6.50	5.64
Speckle noise ($v = 0.01$)	31.99	26.47	20.83	**16.86**
Speckle noise ($v = 0.05$)	40.01	31.70	27.12	**22.49**
Speckle noise ($v = 0.1$)	42.61	36.94	29.58	**24.27**
Median filtering ($w = 3\times3$)	16.54	9.69	8.38	**5.41**
Median filtering ($w = 5\times5$)	25.01	16.88	13.80	**8.89**
Mean filtering ($w = 3\times3$)	13.18	8.74	7.70	**4.03**
Mean filtering ($w = 5\times5$)	22.64	15.80	12.73	**8.46**
Gaussian filtering ($w = 3\times3$)	4.01	3.14	2.01	**1.52**
Gaussian filtering ($w = 5\times5$)	13.13	8.96	7.25	**4.24**
Histogram equalization	4.78	3.96	**2.40**	4.25

processing attacks in comparison with other approaches in the literature. This proves that the eigen decomposition can successfully be used in information hiding field.

To illustrate the difference between coverless information hiding approaches and other traditional approaches, we compare our proposed approach to three other non-coverless information hiding approaches (Sahu and Swain [11], Muhuri et al. [9], and Sahu and Swain [12]) in terms of Peak Signal to Noise Ratio [9] (PSNR) and Structure Similarity Index [9] (SSIM). The range of the SSIM is between -1 and 1. The higher the value of the PSNR and

Table 7 The Resilience (%) of different coverless information hiding approaches

Attack	SIFT+BOF [14]	DCT+LDA [16]	Faster-RCCN [18]	Our proposed
Gaussian noise ($v = 0.001$)	48.00	46.56	**7.23**	10.39
Gaussian noise ($v = 0.005$)	47.97	45.46	18.94	**17.77**
Salt-pepper noise ($r = 0.001$)	48.70	14.07	10.00	**1.79**
Salt-pepper noise ($r = 0.005$)	47.40	25.75	14.37	**5.64**
Speckle noise ($v = 0.01$)	47.05	28.74	**13.12**	16.86
Speckle noise ($v = 0.05$)	47.87	45.36	**21.19**	22.49
Speckle noise ($v = 0.1$)	47.20	59.82	**21.38**	24.27
Median filtering ($w = 3\times3$)	47.99	15.42	15.12	**5.41**
Median filtering ($w = 5\times5$)	49.19	23.35	23.50	**8.89**
Mean filtering ($w = 3\times3$)	49.51	16.32	13.44	**4.03**
Mean filtering ($w = 5\times5$)	49.19	24.40	27.83	**8.46**
Gaussian filtering ($w = 3\times3$)	–	15.57	–	**1.52**
Gaussian filtering ($w = 5\times5$)	–	18.71	–	**4.24**
Histogram equalization	–	29.79	–	**4.25**

SSIM, the higher the similarity between the stego and the cover image. Table 8 displays the best PSNR and SSIM values obtained by the comparing methods. As can be noted that our approach has the highest values since there is no modification has been made to the cover image.

5.4 The information hiding capacity

The proposed approach can store 8 bits of information in each image block. Therefore, the number of image blocks and the amount of overlap between blocks in the cover images can determine the hiding capacity of the proposed approach. In general, as the overlap increases or the size of the block decreases, the number of blocks and the hiding capacity increases. However, when the hiding capacity increases, the resilience of the approach decreases. In the proposed approach, the best results are achieved with the block size $L \times L = 18 \times 18$, the corresponding sub-block size $H \times H = 6 \times 6$, and the overlap is two-third the block width. The hiding capacity (C) can be calculated as

$$C = \begin{cases} 8 \times (\frac{M}{H} - 2) \times (\frac{N}{H} - 2) & \text{For overlapping blocks} \\ 8 \times \frac{M}{L} \times \frac{N}{L} & \text{For non-overlapping blocks} \end{cases} \quad (11)$$

Consider a grayscale image of size 512×512, the hiding capacity of the proposed approach is calculated in Table 9. This Table illustrates the information hiding capacity of different approaches. It can be depicted that our approach has a higher hiding capacity in comparison with other approaches in the literature even when non-overlapping blocks are used for embedding.

5.5 Resistance to detection tools

Many information hiding approaches change cover image contents in order to hide secret messages. Thus, the detection tools can possibly detect the modification in the image by tracing any sign of alterations. Contrarily, our proposed approach depends on mapping relationships between the block hash codes and the message characters. No modification has been made to the cover image contents. Therefore, our approach has a high resistance to existing detection tools.

On the other hand, if an attacker could compromise the communication channel and obtain the location information file and the stego image, the value of other parameters (i.e. block size, sub-block size, the arrangement of neighboring sub-blocks, and the comparison method using PCA) should also be known to calculate the hash code from each block. In this case, the number of different attempts to guess the hash codes (or message characters)

Table 8 Comparison of coverless and traditional information hiding approaches

Approach	PSNR	SSIM
Our proposed	∞	**1**
Sahu and Swain [11]	51.250	0.999
Muhuri et al. [9]	51.668	0.998
Sahu and Swain [12]	48.200	0.997

Table 9 The hiding capacity of different information hiding approaches

Method	Capacity (bits/image)
HOGs [22]	8
SIFT+BOF [14]	8
SIFT [17]	18
DCT+LDA [16]	1-15
faster-RCNN [18]	20 and 25
Our proposed (non-overlapping)	6272
Our proposed (overlapping)	55,112

according to the brute force attack can be calculated using Nt. Suppose the attacker knows the location information file, the Nt can be calculated as

$$Nt = Fs \times \frac{Bs!}{(Bs - Z)!} \tag{12}$$

where Fs is the number of message characters, Bs is the block size in pixels, and Z is the length of hash code in bits. For example, if $Bs = 18 \times 18 = 324$, $Z = 8$, and $Fs = 10$, then $Nt = 11.131 \times 10^{20}$ possible attempts. This means that the attacker should try to extract 8 bits from each target block of size 324. The computations would be more complicated if the location information file is not known. Thus, we have illustrated the security power of the proposed approach.

6 Conclusions

This paper explores the effectiveness of using eigen decomposition for coverless information hiding. Unlike other coverless information hiding approaches, the proposed approach uses only one cover image for embedding a secret message by dividing the image into partially overlapping blocks. The hiding process is accomplished by matching hash codes extracted from image blocks with the secret message characters using mapping relationships. The approach depends on three parameters: overlapping strategy, arrangement of neighboring sub-blocks, and block size. The overlapping and non-overlapping strategy significantly influences the number of unique hash codes calculated from a single image. When the blocks are not overlapped and large block size is used, the number of unique hash codes generated significantly decreases and it would be impossible to conceal a secret message within a single image. On the other hand, the arrangement of neighboring sub-blocks has a lower effect on the number of unique hash codes as compared to the overlapping strategy. The size of the image blocks can control the hiding capacity of the proposed approach. As the block size increases the hiding capacity decreases while the resilience against image processing attacks improves. Compared with other approaches, our approach has higher information hiding capacity and better resilience against image processing attacks. Finally, the approach has a short execution time and it has a better resistance to detection tools since no alteration has been made to the cover image.

Acknowledgements The author would like to thank Mustansiriyah University / Baghdad / Iraq for its support in the present work.

Declarations

Conflict of Interests The author declares that she has no conflict of interest.

References

1. Bishop CM (2006) Pattern recognition and machine learning. Springer
2. Fridrich J, Kodovsky J (2012) Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security 7(3):868–882
3. Guo Y, Li C, Liu Q (2019) R2n: a novel deep learning architecture for rain removal from single image. CMC-Computers Materials and Continua 58(3):829–843
4. Hu D, Wang L, Jiang W, Zheng S, Li B (2018) A novel image steganography method via deep convolutional generative adversarial networks. IEEE Access 6:38303–38314
5. Huang F, Huang J, S Y-Q (2012) New channel selection rule for jpeg steganography. IEEE Transactions on Information Forensics and Security 7(4):1181–1191
6. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on lsb matching revisited. IEEE Transactions on information forensics and security 5(2):201–214
7. Luo Y, Qin J, Xiang X, Tan Y, Liu Q, Xiang L (2020) Coverless real-time image information hiding based on image block matching and dense convolutional network. J Real-Time Image Proc 17(1):125–135
8. McKeon RT (2007) Strange fourier steganography in movies. In: IEEE international conference on electro/information technology, pp 178–182
9. Muhuri PK, Ashraf Z, Goel S (2020) A novel image steganographic method based on integer wavelet transformation and particle swarm optimization. Appl Soft Comput 92:1–26
10. Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: International conference on the theory and applications of cryptographic techniques. Springer, pp 223–238
11. Sahu AK, Swain G (2019) A novel n-rightmost bit replacement image steganography technique. 3D Research 10(2):1–18
12. Sahu AK, Swain G (2020) Reversible image steganography using dual-layer lsb matching. Sensing and Imaging 21(1):1–21
13. Wu D-C, Tsai W-H (2003) A steganographic method for images by pixel-value differencing. Pattern Recogn Lett 24(9-10):1613–1626
14. Yuan C, Xia Z, Sun X (2017) Coverless image steganography based on sift and bof. Journal of Internet Technology 18(2):435–442
15. Zhang X, Wang S (2005) Steganography using multiple-base notational system and human vision sensitivity. IEEE Signal Processing Letters 12(1):67–70
16. Zhang X, Peng F, Long M (2018) Robust coverless image steganography based on dct and lda topic classification. IEEE Transactions on Multimedia 20(12):3223–3238
17. Zheng S, Wang L, Ling B, Hu D (2017) Coverless information hiding based on robust image hashing. In: International conference on intelligent computing (ICIC). Springer, pp 536–547
18. Zhili Z, Cao Y, Wang M (2019) Faster-rcnn based robust coverless information hiding system in cloud environment. IEEE Access
19. Zhiwei K, Jing L, Yigang H (2007) Steganography based on wavelet transform and modulus function. J Syst Eng Electron 18(3):628–632
20. Zhou Z, Sun H, Harit R, Chen X, Sun X (2015) Coverless image steganography without embedding. In: International conference on cloud computing and security. Springer, pp 123–132
21. Zhou Z, Cao Y, Sun X (2016) Coverless information hiding based on bag-of-words model of image. J Appl Sci 34(5):527–536
22. Zhou Z, Wu QJ, Yang CN, Sun X, Pan Z (2017) Coverless image steganography using histograms of oriented gradients-based hashing algorithm. Journal of Internet Technology 18(15):1177–1184
23. Zhou Z, Wu Q, Sun X (2018) Encoding multiple contextual clues for partial-duplicate image retrieval. Pattern Recogn Lett 109:18–26
24. Zhou Z, Mu Y, Wu Q (2019) Coverless image steganography using partial-duplicate image retrieval. Soft Comput 23(13):4927–4938
25. Zou L, Sun J, Gao M, Wan W, Gupta BB (2019) A novel coverless information hiding method based on the average pixel value of the sub-images. Multimedia Tools and Applications 78(7):7965–7980