Coverless Steganography Based on Low Similarity Feature Selection in DCT Domain

Lina TAN [1,2], Jiajun LIU [1], Yu ZHOU [1], Rongyuan CHEN [3]

[1] School of Computer Science at Hunan University of Technology and Business, Changsha 410205, China
[2] School of Computer Science and Electronic Engineering at University of Essex, Colchester CO4 3SQ, UK
[3] School of Resource and Environment at Hunan University of Technology and Business, Changsha 410205, China

ljj20191005@163.com

Submitted April 8, 2023 / Accepted September 8, 2023 / Online first November 13, 2023

Abstract. *Coverless image steganography typically extracts feature sequences from cover images to map information. Once the extracted features have high similarity, it is challenging to construct a complete mapping sequence set, which places a heavy burden on the underlying storage and computation. In order to improve database utilization while increasing the data-hiding capacity, we propose a coverless steganography model based on low-similarity feature selection in the DCT domain. A mapping algorithm is presented based on an 8000-dimensional feature termed CS-DCTR extracted from each image to convert into binary sequences. The high feature dimension leads to a high capacity, ranging from 8 to 25 bits per image. Furthermore, scrambling is employed for feature mapping before building an inverted index tree, considerably enhancing security against steganalysis. Experimental results show that CS-DCTR features exhibit high diversity, averaging 49.3% complete mapping sequences, which indicates lower similarity among CS-DCTR features. The technique also demonstrates resistance to normal operations and benign attacks. The information extraction accuracy rises to 96.7% on average under typical noise attacks. Moreover, our technique achieves excellent performance in terms of hiding capacity, image utilization, and transmission security.*

Keywords

Coverless, steganography, feature collision, DCTR, JPEG

1. Introduction

With the development of information security technology, researchers have proposed many steganography models for image, audio, and video files. Among these approaches, image steganography methods have been mature and widely used, such as [1–5]. Another type of steganography technique [6], [7], known as reversible data hiding, offers a way to recover the original images, making it suitable for medical and military applications. However, all embedding-based models convey the secret message by manipulating some characteristics of cover images. By modifying the cover images, they make themselves easily suspectable to attackers. Correspondingly, steganalysis for determining whether an image contains secret information has been widely concerned and developed. Besides some traditional methods, such as [8–10], there are some deep learning-based schemes, such as Xu-Net [11] and SR-Net [12].

Considering the aforementioned statistical steganalysis issue, Zhou et al. [13] proposed a coverless steganography scheme without any modifications to the cover images. State-of-the-art steganalysis techniques are ineffective against this approach. To further improve the robustness against typical attacks, they developed a novel method [14] in 2016 that uses image features to extract visual keywords based on the Bag-of-Words (BOW) model. Zheng et al. [15] applied robust image hashing based on orientation information of Scale-Invariant Feature Transform (SIFT) [16] feature points for coverless hiding, which can withstand attacks involving rotation and scaling. Moreover, it doubles the hiding capacity compared to the method in [13]. Cao et al. [17] used the molecular structure images of material (MSIM) as the cover images, which have a more general average pixel value. Experiments proved its superiority in hiding capacity. Later, Zhang et al. [18] introduced a discrete cosine transform (DCT) steganography model that creates a robust feature sequence by computing the DCT coefficients of adjacent sub-blocks segmented from an image. Similarly, Liu et al. [19] employed the discrete wavelet transform (DWT) coefficients between the blocks of zigzag scan to generate feature sequences, which exhibit superior performance in dealing with abrupt signals compared to the DCT. The problem shared by these two approaches is that robustness decreases as the number of sub-blocks increases. Although these methods are more resistant to typical noise attacks than spatial domain-based ones, they are more vulnerable to geometric attacks. Luo et al. [20], therefore, applied deep learning in coverless steganography by utilizing Faster Region-Based Convolutional Neural Networks (Faster

DOI: 10.13164/re.2023.0603

Method	Advantages	Disadvantages
[13], [14]	Simple functional implementation	Small hiding capacity and poor robustness
[15]	Stronger robustness against rotation & scaling attacks, improved hiding capacity	High computational complexity
[17]	Further increased capacity	Limited by molecular structure images of material (MSIM)
[18]	More robust against noise attacks	Fragile to geometric attacks
[19]	Ability to handle abrupt signals better than the DCT used in [18]	Fragile to geometric attacks
[20–23]	More robust against geometric attacks	Less robust against noise attacks compared to frequency domain-based methods
[24, 25]	Less database load	Small hiding capacity, unnatural image generation, low detection accuracy
[26]	Increased security with styles transfer	Low extraction accuracy
[7]	Higher security and robustness	Small hiding capacity
Ours	Low collision rate of the feature sequences and the minimal burden on the database	A slightly lower robustness

Tab. 1. Comparison of coverless steganography.

RCNN) to detect objects in cover images. This multi-object identification produces robust binary sequences that increase robustness against geometric attacks. Other deep learning-based coverless steganography exists as well, like [21–23]. These techniques are robust to geometric attacks but less tolerant to noise. Furthermore, Liu et al. [24] and Hu et al. [25] proposed coverless schemes based on Auxiliary Classifier Generative Adversarial Network (ACGAN) and Deep Convolutional Generative Adversarial Network (DCGAN), respectively. They create stego images using the GAN network, which can lessen the database load. However, the generated images are not realistic and natural enough. Based on [25], Zhang et al. [26] employ CycleGAN to further enhance security by adding style transfer to the generated images. Later, Chen et al. [27] generated stego images using StarGAN [28] based on the mapping relationship between facial attributes and secret information. Despite its small capacity, it is extremely secure and robust. Table 1 displays the benefits and drawbacks of the aforementioned coverless steganography based on mapping rules.

In theory, a perfect mapping of secret information is possible if a large enough number of images are maintained in the database for coverless steganography. However, features extracted from certain images based on existing models generally share high similarities. Since these images produce identical feature sequences, it is challenging to build a complete image database. The image database burden increases dramatically whenever capacity requirements increase slightly. That is also one of the main reasons for the limited capacity of coverless steganography.

To improve steganography capacity and image retrieval efficiency, we propose a coverless scheme based on low similarity feature selection. An iterative algorithm is developed for a high-dimensional rich model based on low-similarity CS-DCTR features, which exploit the first-order statistics of quantized noise residuals obtained from decompressed JPEG images.

The main contributions of this paper are as follows:

1. Accounting for the fact that the collision rate of mapping sequences would put significant stress on the database, we develop a recombination technique on the submatrices of residuals obtained from the basic patterns in the DCT. A new mapping rule is designed to considerably reduce the collision rate of feature sequences, thus boosting the matching efficiency.

2. In order to improve the security of anti-steganalysis and lower the transmission load, a scrambling program is applied to the sequences after feature mapping. We issue a unique scramble key to each recipient as the same seed key used for scrambling. After scrambling, the pseudo-random distribution of sequence values provides better uniform quantization, increasing the diversity of feature mapping.

The rest of this paper is organized as follows. Section 2 gives the related knowledge and basic tools applied in the scheme; Section 3 introduces the main idea of the proposed method; Section 4 provides the experimental results and analysis; Section 5 is the conclusion, where we discuss future directions.

2. Preliminary

2.1 Notation

The idea of DCTR was first proposed in [29]. We propose a coverless image steganography method based on the modified version of DCTR. The notations provided in Tab. 2 are applied to calculate the CS-DCTR feature.

2.2 DCTR Model

DCTR is the first-order statistics of the quantization noise residual obtained after applying the DCT to the decompressed JPEG image. It also can be regarded as a projection model in the DCT domain. This feature sets have low dimensionality and computational complexity. The extraction process of DCTR features is shown in Fig. 1. We first perform an inverse transformation on a selected JPEG image from the DCT domain to the spatial domain and then

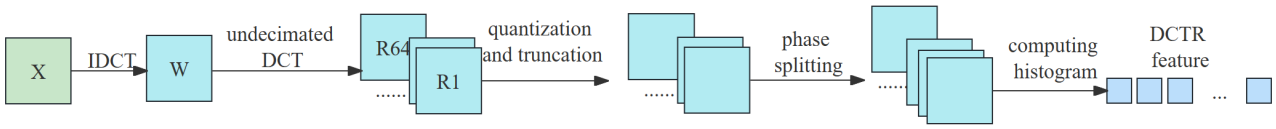

Fig. 1. Generation process of DCTR components.

Notations	Descriptions
$\mathbf{W}$	spatial matrix of an image
$\mathbf{X}$	DCT coefficient matrix of an image
x_{uv}	elements of X
w_{ij}	elements of W
SM	secret message
C	cover images
Iv	inverted index
$S = s_1 \| \| s_2 \| \| ... \| \| s_m$	segmented secret information
$SI = si_1 \| \| si_2 \| \| ... \| \| si_m$	the set of stego images
$F = f_1 \| \| f_2 \| \| ... \| \| f_m$	the CS-DCTR feature vector extracted from a stego image
$SQ = sq_1 \| \| sq_2 \| \| ... \| \| sq_m$	all sequences mapped from F
$sq_i = sq_i^1 \| \| sq_i^2 \| \| ... \| \| sq_i^{64}$	64 feature sequences extracted from the ith stego image
$Bs = bs_1 \| \| bs_2 \| \| ... \| \| bs_m$	the binary sequence before scrambling
$SKey = skey_1 \| \| skey_2 \| \| ... \| \| skey_e$	shuffling keys held by receivers
$PKey = pkey_1 \| \| pkey_2 \| \| ... \| \| pkey_m$	keys with position information

Tab. 2. Notation.

compute the undecimated DCT to obtain its residual matrix. The matrix coefficients are further truncated and quantized to obtain a fixed range of values. Thereafter, the processed matrix is separated and merged into multiple sub-matrices. Finally, its histogram is calculated.

3. Proposed Work

3.1 CS-DCTR Feature Extraction

For coverless steganography, a cover image, which is also a natural image, can generate a hidden bitstream according to the feature mapping rule. This model is inherently resistant to steganalysis due to its virtue of not modifying the content of stego images in any way. We developed the CS-DCTR, a modified version of DCTR suggested in [29]. The algorithm involves the following steps.

3.1.1 Blockwise IDCT

For JPEG images, their spatial-domain data are first decompressed without quantization based on inverse discrete cosine transform (IDCT) before computing the undecimated DCTR features. The 8×8 blockwise DCT coefficient matrix $\mathbf{X}$ of a given image is fed into the IDCT algorithm, creating an output spatial matrix $\mathbf{W}$, whose element w_{ij} is given by

$$w_{ij} = \sum_{i=0}^{7} \sum_{j=0}^{7} x_{uv} \cos \frac{(2i+1)\pi}{16} \cos \frac{(2j+1)\pi}{16} \tag{1}$$

where x_{uv} is the element of 8×8 blocks of DCT coefficient matrix $\mathbf{X}$. c is a parameter whose value is:

$$c_u = \begin{cases} \frac{\sqrt{2}}{4}, & \text{if } u = 0; \\ \frac{1}{2}, & \text{if } u \neq 0. \end{cases} \tag{2}$$

It can also be expressed as

$$\mathbf{W}_8 = \mathbf{A}^{\mathrm{T}} \mathbf{X}_8 \mathbf{A}. \tag{3}$$

where $\mathbf{A}$ is the cosine coefficient matrix.

3.1.2 Undecimated DCT

For a grayscale image $\mathbf{X}$ with size 256×256, there are 64 convolution kernels $\mathbf{B}^{k,l}$ to compute the undecimated DCT coefficients. The size of a convolution kernel is 8×8. The kernel corresponding to the position (k, l) in the DCT domain is denoted as $\mathbf{B}^{k,l}$, which can be calculated by

$$\mathbf{B}^{k,l} = {\mathbf{B}_{mn}}^{k,l}, \quad 0 \leq m, n \leq 7, \tag{4}$$

$${\mathbf{B}_{mn}}^{k,l} = \frac{\mathbf{W}_k \mathbf{W}_l}{4} \cos \frac{\pi k(2m+1)}{16} \cos \frac{\pi l(2n+1)}{16} \tag{5}$$

where $\mathbf{W}_0 = \frac{1}{\sqrt{2}}$,$\mathbf{W}_k = 1$ for $k > 0$.

After all convolution kernel parameters are defined, they are used to filter residuals. The calculation formulas are determined below.

$$U(\mathbf{X}) = \{\mathbf{R}^{k,l} | 0 \leq k, l \leq 7\}, \tag{6}$$

$$\mathbf{R}^{k,l} = \mathbf{X} * \mathbf{B}^{k,l}. \tag{7}$$

where $*$ denotes convolution operation. In order to reduce the collision rate of features, the stride of the convolution operation is set to 8. After filtering the original image matrix $\mathbf{R}^{k,l}$, we get 64 residual matrices with size 32×32.

3.1.3 Quantization and Truncation

To improve the efficiency of computing histogram features in DCTR, all residual matrix values must be kept within a fixed range. The operation consists of three steps, first quantizing all elements of the matrices, then rounding quantized matrices, and finally changing the element values to the range $[-t, t]$ using the truncation function. In order to enhance the precision of the results, we refrain from rounding the elements, thereby maintaining their originality. The operation applied in CS-DCTR is defined as

$$r' = \mathrm{trunc}_t\left(\frac{|r|}{q}\right) \tag{8}$$

where r and r' are the input and output parameters, respectively. t is set to $[0, 4]$, which is the same as DCTR. The value of q depends on the JPEG quantization factor Q, which is formulated as (9). Since quantization factors Q of all images used in this paper are 75, the q is 4.

$$q = 8 \times \left(2 - \frac{Q}{50}\right), \quad Q \in 50, 51, \cdots, 99. \tag{9}$$

3.1.4 Recombination

In DCTR, each residual matrix is split and merged in this procedure based on the phase position. Due to the 8×8 size of the cosine coefficient matrix for computing DCT, an undecimated DCT matrix will be split into $8 \times 8 = 64$ submatrices. The operation is given as follows:

$$\mathbf{U}^{k,l}_{a,b}(i, j) = \mathbf{U}^{k,l}(i + 8 \times a, j + 8 \times b) \tag{10}$$

where $\mathbf{U}^{k,l}$ is the undecimated DCT matrix corresponding to the frequency (k, l), and the submatrix whose elements at position (a, b) in the DCT block is denoted as $\mathbf{U}^{k,l}_{a,b}$. The parameters $k, l, a, b, i, j \in \{0, 1, 2, \cdots, 7\}$ are applied here.

And before creating the histograms, the submatrices need to be merged further. As mentioned earlier, an undecimated DCT matrix can be split and merged into 64 submatrices. According to the symmetrical properties of projection vectors, these 64 submatrices can be merged into 25 phase matrices. Figure 2 describes the merging process. There is a grid of 8×8 representing 64 phases in Fig. 2, where the same phases are denoted by identical letters and colors. A derived 5-bin histogram feature from the final merged matrix follows the phase-based merging technique. The split of an undecimated DCT matrix into 25 merged phase matrices allows for the extraction of a histogram feature with $25 \times 5 = 125$ bins. Then $64 \times 125 = 8000$ dimensional histogram feature can be extracted using the 64 residual matrices.

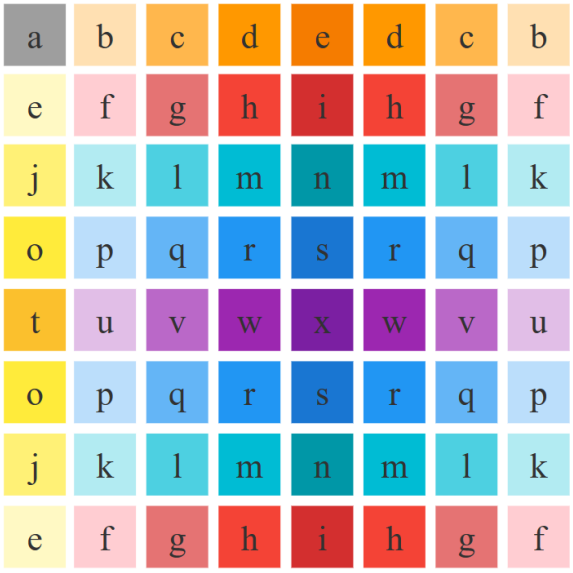

Fig. 2. Diagram of phase matrix merging.

However, in our scheme, a residual matrix with a size of 32×32 is obtained by each convolution kernel. After experimental comparison, we selected some optimal coefficients for recombination to reduce the similarity of features. Because in CS-DCTR, the convolution stride is set to 8, its symmetry no longer exists. But the original algorithm of merging submatrices in DCTR can be used to reorganize the residual matrix. At the same time, we combine the original operation of merging submatrix with the operation of phase splitting and merging to obtain our reorganization algorithm.

Figure 3 describes the process of recombination. A residual matrix is divided into 64 submatrices with the size of 4×4. The 64 submatrices can be merged into 25 submatrices. There is an 8×8 grid, which represents the 64 unmerged submatrices. Submatrices with the same letters represent those who can merge, and accordingly 64 submatrices are synthesized into 25 ones.

3.1.5 Gaussian Histogram

For the same reason as in Step 3 and as motivated by [30], we consider substituting a Gaussian histogram for the traditional histogram. The cumulant of the Gaussian function is used to model the histogram bins in DCTR, which is calculated from

$$h_g(c) = \frac{1}{\| X \|} \sum_{i,j} \mathrm{e}^{-\frac{(x_{ij}-c)^2}{\sigma^2}} \tag{11}$$

where σ is a parameter that determines the shape of the Gaussian function, and its recommended value is 0.6 in both [30] and [31] .

3.2 Framework

The initial step of this scheme is to allocate various shuffling keys to each receiver. After the CS-DCTR feature extraction of all cover images in the database, they are

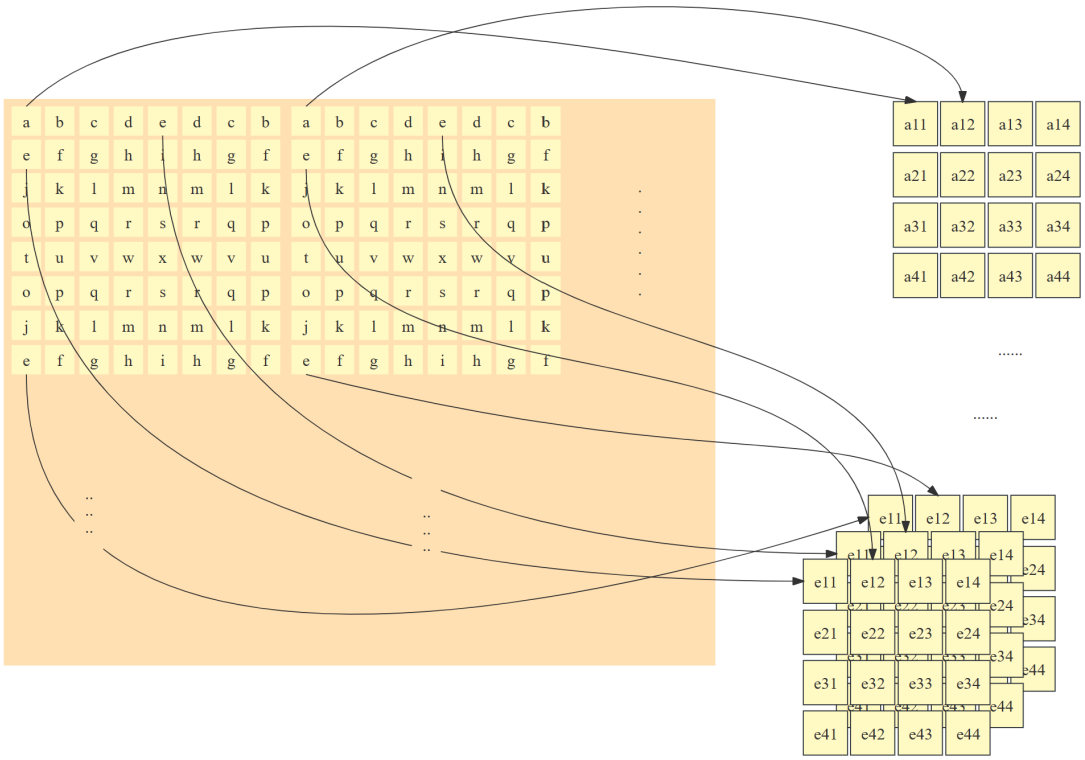

Fig. 3. The recombination process of residual matrices.

converted into binary sequences by a mapping algorithm. Different scrambled sequences, one for each receiver, are then produced using the shuffling keys. Once the appropriate images are chosen from the inverted indices to serve as the stego ones, they are sent directly to the recipients with the position keys without being altered throughout the steganographic process. After acquiring the stego images, the receiver computes the CS-DCTR feature of each image in turn and extracts the correlative binary sequence according to the position key. The retrieved sequences are subsequently subjected to the scrambling procedure using his specific shuffling key. Concatenating the scrambled sequences for each image will finally reveal the secret message. Figure 4 depicts the entire process of information hiding and extraction.

If a stego image is expected to generate an n-bit binary string, in theory, an image database for a coverless scheme requires at least 2^n images. In other words, the more bits of information you try to hide, the larger your image database will be. Receivers should be issued their specific shuffling keys in advance so they can accurately decipher the secret message.

3.3 Preprocessing

Under this system, the sender allocates a unique shuffling key in complete secrecy for each receiver previously. Before transmitting the secret information, extra preprocessing work is required, such as creating scrambled sequences, establishing inverted indexes, and so on.

3.3.1 Feature Sequence Mapping

The time cost will significantly increase due to potential repetitive extraction operations if feature sequences are only extracted from those images when they are selected as stego ones each time. Thus, it makes sense to map all the images in the database to binary sequences in advance to save time.

The joint determination of 64 submatrices yields an 8000-dimensional CS-DCTR feature. Accordingly, if the feature vector extracted from a submatrix is converted to a binary sequence via a mapping approach, each image will be mapped to 64 binary sequences. It is an effective way to reduce the computational burden.

For the 125-dimensional feature vector derived from each submatrix, we divide it into 25 intervals by summing five successive bins, which are denoted as $\mathrm{sum}[i], 1 \le i \le 25$. A sequence of 25 bits will be produced by the formula below. As the longer the sequence is, the larger the database needs to be built. Considering the number of images, the value of steganography capacity n can be determined by the user, that is, the first n bits of the sequence are used to hide information. In our approach, n is recommended to $[8, \ldots, 25]$.

$$\begin{cases} sq[i] = \text{'1'}, & \text{if} \quad \mathrm{sum}[i] \ge ave; \\ sq[i] = \text{'0'}, & \text{if} \quad \mathrm{sum}[i] < ave; \end{cases} \tag{12}$$

where $sq[i]$ represents the ith bit of the binary sequence, and ave denotes the mean of sum. It is worth noting that a carrier image can theoretically be mapped into 64 feature sequences, some of which may be identical yet, as illustrated in Fig. 5. These redundant duplicate sequences will be removed.

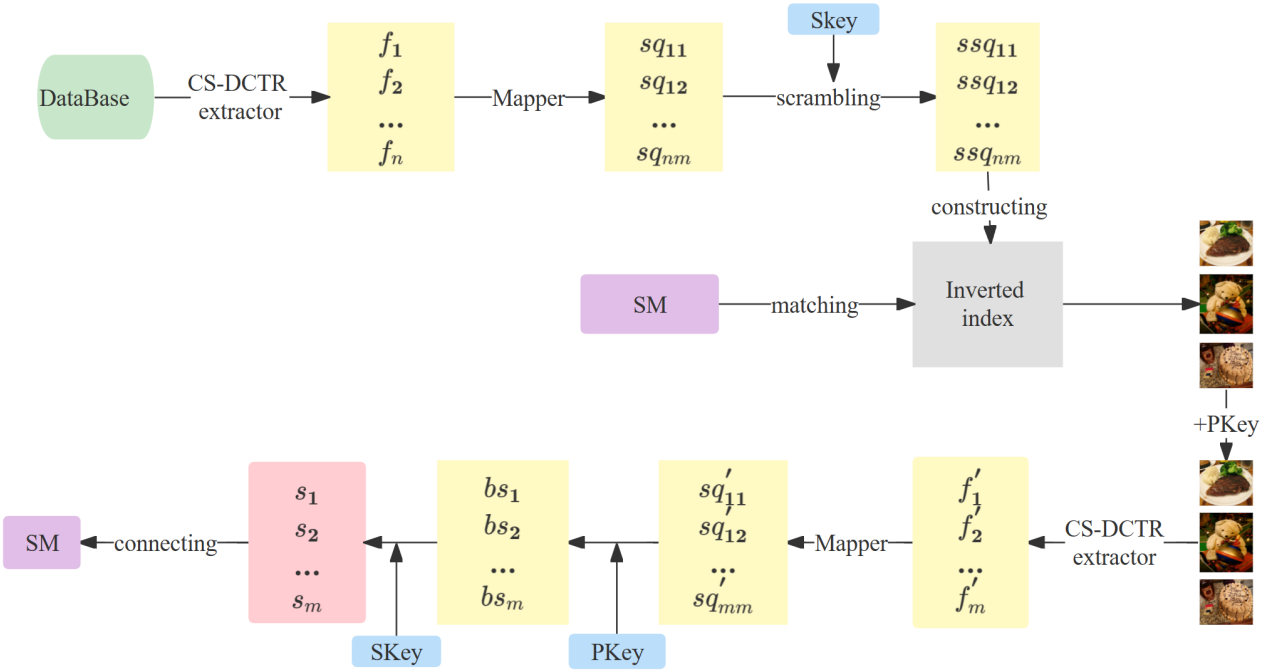

Fig. 4. Block diagram of the proposed coverless image steganography.

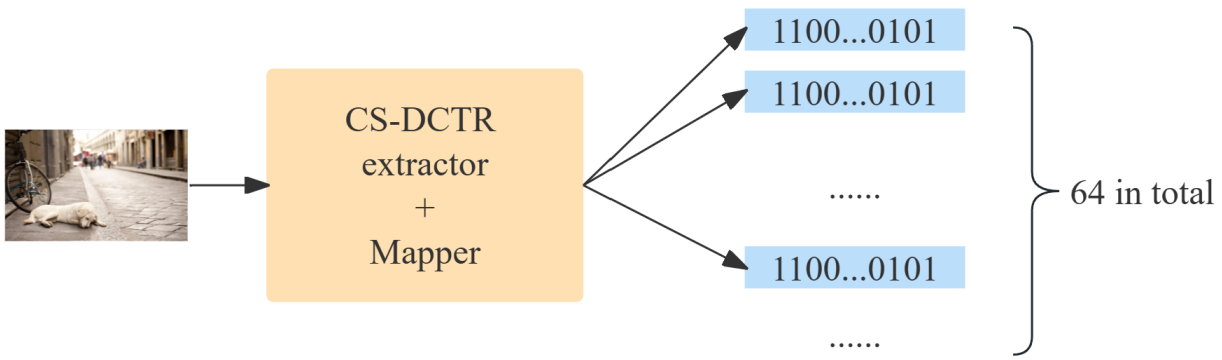

Fig. 5. Binary sequence mapping process.

3.3.2 Shuffling Key Distribution

There are many algorithms for shuffling a sequence, and after considering the randomness and efficiency of the algorithm, the Fisher-Yates Shuffle algorithm is used. Ronald A. Fisher and Frank Yates were the first to suggest the shuffling approach known as the Fisher-Yates Shuffle. The basic idea is to randomly select a number that has not been taken before from the original array to the new array.

Each receiver uses a unique shuffling key to scramble the signature sequence in the process. After extracting the CS-DCTR feature from each cover image, different scrambled sequences will be produced using various shuffling keys.There are two ways to provide a user his shuffling key: one is to issue the key to each recipient manually, and the other is to enter the user ID into a random function to generate a shuffling key specifically for the user. Since it is easier for an attacker to guess the user ID using the exhaustive method, we consider the first solution more secure.

3.3.3 Inverted Index Construction

To improve the search efficiency of cover images, we designed a three-level inverted index tree Iv for each receiver according to their shuffling key. The final scrambled sequences $\{ssq_{ij}\}, i, j \in \mathbb{Z}$ obtained in the previous stage make up the first-layer elements of Iv. The second-layer components are the binary sequences $\{sq_{ij}\}, i, j \in \mathbb{Z}$ obtained after deduplicating the feature sequence mapping results, i.e., before using the scrambling technique. The third layer comprises the order numbers of the feature mapping sequences, and each position subkey $pkey_i, i \in \mathbb{Z}$ corresponds to an image list. Some feature sequences with the same order number of various images might be identical. Images mapped to identical binary sequences at the same ordinal positions according to (12) will be put into the same list of subkeys. In order to achieve the complete matching of all binary bits of the hidden data, the first layer should be a closure set of n-bit binary sequence shift operations, i.e., the number of root nodes is 2^n. Therefore, it is necessary to ensure the diversity of the image library. Figure 6 shows the inverted index structure.

3.4 Coverless Image Steganography

As discussed in previous sections, every receiver has a unique shuffle key. For the sake of clarity, we use the example of a receiver named Jack in this part. The same procedures apply to other receivers. The steganography process is taken by selecting cover images based on the inverted index tree, which consists of the following steps.

1. The sender first converts the secret message into a binary string and divides it into n-bit information segments. If the length of the binary message is not divisible by n, an appropriate number of '0's will be added to the higher bits of the binary string. Thus, we can obtain a binary string S shown below for the secret information SM of length L.

$$S = s_1||s_2||...||s_m \tag{13}$$

where m is represented by

$$m = \begin{cases} L/n, & L/n = 0 \\ L/n + 1, & \text{else} \end{cases} \tag{14}$$

2. For the last information segment s_m of S, we find its equivalent sequence from the first-level index of the inverted index structure. After that, we trace down a branch of the root node previously identified at the first layer until reaching the leaf layer, from which we pick out an arbitrary position subkey $pkey_i, i \in \mathbb{Z}$ to assign to Jack. If there is a multi-branch scenario during the search process, we can freely choose a path based on a random seed until we approach the leaf layer. More than one image may be found in the image list corresponding to the subkey $pkey_i$. In this case, the sender can randomly choose an image that hasn't been used in this round of steganography from the image list and add it to the cover image collection:

$$si_m = \text{Match}(s_m, Iv). \tag{15}$$

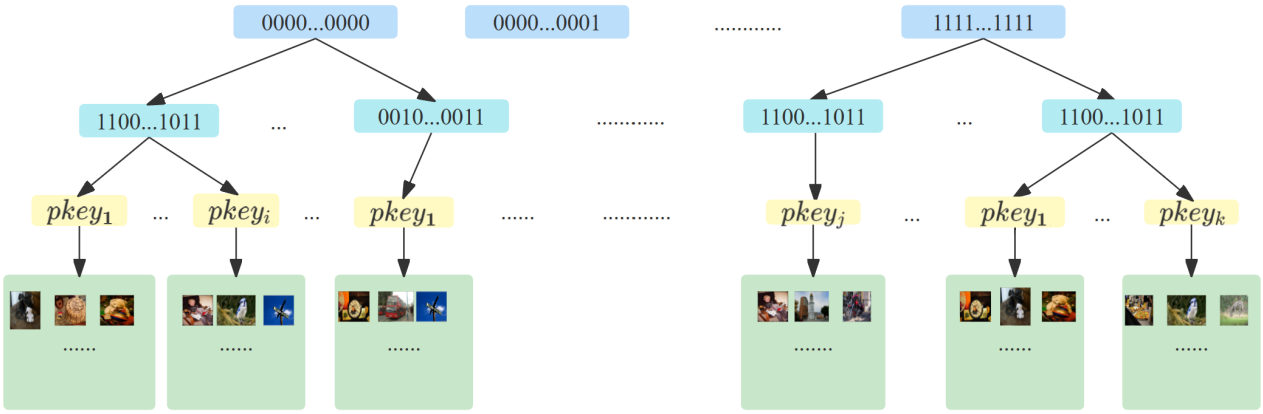

Fig. 6. Inverted index tree Iv, with $1 \le i, j, k \le 64$ for the third layer.

3. For the segments s_{m-1} to s_1 of S in turn, repeat Step 2 until all the secret information segments are successfully matched to a stego image. Then a stego image set is obtained:

$$SI = si_1||si_2||...||si_m. \tag{16}$$

4. All subkeys $pkey_i$ acquired during the loop procedure must be recorded to create the position key $PKey$, which must be sent to Jack along with the stego image set:

$$PKey = pkey_1||pkey_2||...||pkey_m. \tag{17}$$

For secure data transmission, $PKey$ can be delivered to Jack in a public-key encryption method at a separate time than the stego images. Furthermore, it is possible for various information segments to map to the same stego image during the image matching process. In this case, we usually choose an alternative stego image unused before. If we have no other options in the same image list, we either need to go back to the node of the preceding layer to search for another branch again, or we might consider making the image library more diverse. Algorithm 1 offers a brief explanation of the entire information hiding process.

Algorithm 1. Coverless image steganography.

Require: Image database: C; Secret message : SM; Shuffling key: $Skey$
Ensure: Stego images : $SI = si_1||si_2||...||si_m$; position information : $Pkey = pkey_1||pkey_2||...||pkey_m$
  $Pkey = NULL$
  r = length(C)
  for j from 1 to r **do**
    Extract the CS-DCTR feature: F = Extract(C_j)
    Generate the binary sequence: SQ = Mapping(F)
    Scrambling the sequence via $SKey$: $SQ^{'}$ = Scramble(SQ)
  end for
  Construct the inverted index: Iv = Construct($C, SQ, SQ^{'}$)
  Convert SM to binary string S : S = Convert(SM)
  Divide S : $S = s_1||s_2||...||s_m$
  for j from m downto1 **do**
    Match the stego image si_j : si_j = Match(s_j, Iv)
    Record the position information : $Pkey = pkey_j||Pkey$
  end for
  Connect all stego images : $SI = si_1||si_2||...||si_m$
  return SI and $PKey$

3.5 Information Extraction

After Jack receives the stego image set and $PKey$ via public channels separately, he can extract accurate data. However, some compensatory measures must be considered in light of potential threats such as geometric distortion and additive noise. The geometric matching algorithm uses some calibration information attached to the stego images to return a distance of the position, angle, and bounding rectangle between the reference and the distorted images. A simple Gaussian filter is used to suppress the minor additive noise. The steps of secret information extraction are given as follows.

1. After receiving the stego image set SI, the images are first corrected to restore spatial synchronization. The CS-DCTR feature F are then extracted from the corrected stego images:

$$F = f_1||f_2||...||f_m. \tag{18}$$

2. Each element f_i of F is mapped to 64 binary sequences by the feature mapping algorithm described earlier. SQ represents the concatenation of all sequences mapped from F. And sq_i is the 64 sequence sets mapped from f_i:

$$SQ = sq_1||sq_2||...||sq_m, \tag{19}$$

$$sq_i = sq_i^1||sq_i^2||...||sq_i^{64} \tag{20}$$

where $0 < i \leq m$.

3. For sq_1, select the sequence bs_1 of the corresponding position based on $pkey_1$:

$$bs_1 = \text{Select}(sq_1, pkey_1). \tag{21}$$

4. Scramble bs_1 according to the shuffling key $SKey$ held by Jack. A scrambled sequence s_1 will be obtained, which is the secret information segment hidden in this stego image:

$$s_1 = \text{Scramble}(bs_1, SKey). \tag{22}$$

5. For the segments sq_2 to sq_m and $pkey_2$ to $pkey_m$ in turn, repeat Step 3 until secret message segments are successfully extracted from all stego images. Concatenate all segments of S and remove the extra '0' added on its higher bits to get the final secret message SM:

$$S = s_1||s_2||...||s_m. \tag{23}$$

According to the above description, the data extraction process can be summarized as Algorithm 2.

Algorithm 2. Secret message extraction.

Require: Stego images : $SI = si_1||si_2||...||si_m$; position information : $PKey = pkey_1||pkey_2||...||pkey_m$
Ensure: Secret message : SM
  for j from 1 to m **do**
    Extract the CS-DCTR feature in si_j : f_j = Extract(si_j)
    Generate the binary sequences : $sq_i = sq_i^1||sq_i^2||...||sq_i^{64}$
    Find the correct sequence via $PKey$: bs_j = Select($sq_1, pkey_1$)
    Scramble bs_j via $SKey$: s_j = Scramble($bs_j, SKey$)
  end for
  Contact si to S : $S = s_1||s_2||...||s_m$
  Convert S to SM
  return SM

4. Experiments and Analysis

Experimental environment: AMD R7-4800U CPU @ 1.80 GHz and 16.00 GB RAM. All experiments are completed in Pycharm and MATLAB R2021a.

Data set: BOSSbase 1.01 contains 10,000 grayscale images in PGM format. All images in BOSSbase are set to 512×512 in size. For convenience, some preprocessing steps were taken before conducting the experiments. Firstly, all images in BOSSbase were resized to 256×256 by the cubic linear interpolation algorithm. Furthermore, all images were converted to JPEG format with a quality factor of 75. COCO2017 is a large-scale dataset that contains more than 330000 images, of which 220000 are labeled, with 1.5 million objects, 80 object categories, and 91 stuff categories.

4.1 Hiding Capacity

Most coverless steganography techniques have lower capacity than conventional methods, which is mainly due to the image feature mapping mechanism. In this scheme, the CS-DCTR feature extracted from each image can produce 64 n-bit mapping sequences, and any value from 8 to 25 can be used for n. As a result, the optimal hiding capacity of our approach is n bits per stego image. Nevertheless, a higher hiding rate implies the requirement for a larger image database with more than $2^n/64$ images in theory.

Table 3 displays a capacity comparison with a few state-of-the-art techniques. These techniques include two mapping approaches [18], [19] utilizing image transform-domain coefficients, which best represent the optimal frequency-domain mapping rule, a pixel-feature-based mapping method [13], which represents the conventional rule-based steganography, and a mapping method [20] based on image object category information, which serves as the quintessential semantically-driven mapping rule. Among them, The techniques [13, 18, 19] turn out to perform poorly. The method of [20] enables any object detected in an image to be matched to a 6-bit string, allowing each image to be mapped to a string of $6 \times m$ bits according to the m objects it contains. However, most images in the database only have $1 \sim 3$ detectable objects, resulting in a general hiding capacity of $6 \sim 18$ bits.

As shown in Tab. 3, it is clear that the proposed solution has better hiding capacity under the assumption of affordable image library storage. Moreover, a dynamic parameter n varying from 8 to 25 renders a flexible way as a tradeoff between the storage overhead and transmission security.

Methods	**Capacity** [bits/image]
Zhou's [13]	8
DCT [18]	$1 \sim 15$
DWT [19]	$1 \sim 15$
RCNN [20]	$6 \times m$
Ours	$8 \sim 25$

Tab. 3. The capacity of schemes.

Image number	1000	2000	3000
Inverted index construction efficiency [s/image]	0.020	0.019	0.016

Tab. 4. The efficiency of index contruction.

Image number	400	800	1200
Image matching efficiency [s]	0.063	0.110	0.173
Information extraction efficiency [s]	3.393	3.995	4.694

Tab. 5. The efficiency of images matching.

4.2 Time Efficiency

This section offers an experimental evaluation of the computational performance of the proposed method. The metric analysis demonstrates the algorithm's efficiency in providing fast and reliable stego image processing.

There are three main components that make up the time consumption: 1. Time spent on inverted index construction, which is tested to determine how long it will take to develop the database's index structure. Preprocessing steps such as feature extraction, feature representation, and index construction are parts of the process. 2. The time overhead for image matching, which is examined to measure the time taken to match a given query image with images in the database. The matching speed is assessed by varying the length of the secret information, ensuring a thorough evaluation of the algorithm's computational efficiency. 3. The time cost of information extraction, focusing on the time required to extract the hidden information from the stego images. This covers the decoding process, feature analysis, and information recovery.

From Tabs. 4 and 5, it can be observed that our approach significantly reduces the time required to construct the inverted index, with each image taking no more than 0.02 seconds. This is due to the remarkable efficiency of image feature extraction and binary sequence mapping. Furthermore, information hiding is also incredibly effective owing to the inverted index. However, because of the additional feature extraction process, information extraction takes slightly longer to complete.

4.3 Feature Similarity

Since the feature sequences extracted from some images are identical, it is challenging to build a complete database to enable full mapping of hidden data. Low feature similarity reduces the collision rate of image matching results, which has a beneficial impact on the size of the image database. Here, we evaluate the feature similarity of several approaches based on the number of binary sequences generated from the image sets under various conditions of hiding capacity. Test images were selected from Bossbass 1.01 and COCO2017. The evaluation metric is given by

$$P = \frac{M}{T} \times 100\% \tag{24}$$

Database	Number of image	Capacity [bits/image]	Ours	Zhou's [13]	DCT [18]	DWT [19]
Bossbase	2048	11	**52.2%**	43.4%	45.0%	44.5%
	8192	13	**50.1%**	40.0%	40.7%	41.2%
COCO	2048	11	**50.4%**	45.1%	43.8%	47.1%
	8129	13	**48.3%**	39.7%	40.3%	43.5%
	65536	16	**45.6%**	37.3%	36.2%	38.1%

Tab. 6. Feature similarity of various schemes.

Attack	Parameters	Bossbase	COCO	Pixel [13]	SIFT [15]	DCT [18]
Gaussian noise	The mean u: 0, the variances q: 0.001	94.1%	94.7%	**95.8%**	65.6%	95.4%
Salt and pepper noise	The mean u: 0, the variances q: 0.001	98.7%	98.0%	99.0%	90.8%	**99.2%**
Speckle noise	The mean u: 0, the variances q: 0.01	95.3%	96.4%	**96.6%**	74.4%	96.2%
Median filtering	The window size: 3×3	98.7%	97.1%	**99.6%**	88.4%	99.4%
Mean filtering	The window size: 3×3	94.5%	96.2%	**98.8%**	73.6%	95.8%
Gaussian filtering	The window size: 3×3	96.6%	95.8%	99.8%	92.6%	**100%**
Scaling	The scaling ratios: 3	99.2%	98.6%	99.6%	95.2%	**100%**

Tab. 7. The robustness of attack.

where T is the total number of images in the database, and M is the number of binary sequences successfully extracted from all images. The larger the value of P, the lower the feature similarity is.

The comparison results are shown in Tab. 6. We randomly selected 2048 and 8192 images from Bossbase1.01 in the first group of experiments with a hiding capacity of 11 and 13 bits, respectively. For the second group of experiments, 2048, 8192, and 65536 test images were selected from COCO2017 with hiding rates of 11, 13, and 16 bits per image, respectively. The findings presented in Tab. 6 indicate that, in the case of two different databases, the similarity of CS-DCTR features, which are extracted from our proposed coverless steganography scheme, is lower compared to those obtained from other coverless steganography schemes at different bit capacities.

This observation shows that our CS-DCTR feature effectively minimizes collisions among images and is superior to other coverless steganography schemes. This demonstrates the potential of our method in reducing the database load, as it not only effectively hides secret information but also reduces the computational and storage requirements for databases processing large amounts of image data.

4.4 Robustness

Stego images may be subject to innocent or malicious attacks during transmission, resulting in the loss of image content and destruction of features. To ensure the algorithm's robustness and effectivity, we introduced some typical geometric attacks indicated in Tab. 7.

For most approaches, any change in image pixels may affect the accuracy of information extraction, so the robustness of steganography methods is determined by the accuracy of information extraction from images. The calculation formula is defined as

$$R = \frac{SM_{\mathrm{c}}}{SM_{\mathrm{o}}} \times 100\% \tag{25}$$

where SM_{c} is the length of the correctly extracted secret message, and SM_{o} is the total length of the original message.

Extensive experiments were performed, such as Recursive Set of Transformations (RST) and additive noise, to test the detection accuracy against attacks. In detail, we subjected the images to four types of geometric attacks, including center cropping, edge cropping, rotation, and translation, as well as seven types of noise attacks, including Gaussian noise, salt and pepper noise, speckle noise, median filtering, mean filtering, Gaussian filtering, and scaling. These noise attacks with different parameters are depicted in Fig. 7.

Table 7 summarizes the performance of various attack parameters on Bossbase and COCO, with evaluations conducted on the COCO dataset for the other four methods. Our method achieves relatively good anti-noise performance. General noise attack has little effect on the CS-DCTR features, reflecting its anti-interference ability. However, the effectiveness of our method is relatively low when encountering geometric attacks, which may result in some alterations or loss of some image features, seriously destroying the integrity of CS-DCTR features.

Although the robustness on some metrics is not optimal compared with previous approaches, our method still shows good overall performance. Considering the transmission characteristics of information hiding techniques, our method concentrates study on database storage efficiency while maintaining resilience against typical attacks. This approach achieves a good balance between robustness and resource utilization. Particularly when confronted with noise attacks, our method achieves an average robustness of 96.7% on both Bossbase and COCO.

4.5 Security

Existing steganalysis models identify the presence of secret information by examining changes in the statistical properties of images. Since the coverless image steganography presented in this research does not modify the cover images, existing steganalyzers cannot detect anomalies from

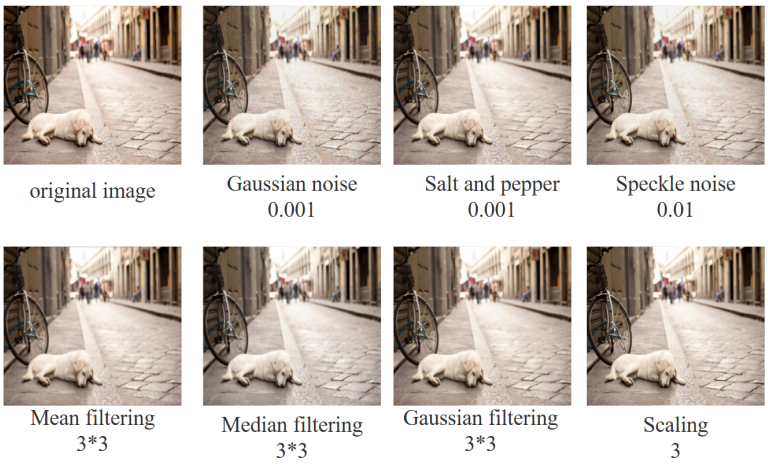

Fig. 7. Various types of noise attack.

stego images. Meanwhile, the outstanding hiding rate performance reduces the image load during transmission, further lowering the risk of raising suspicion in attackers. Moreover, we assign a unique shuffling key to each receiver that needs to be transmitted in advance to address the uneven image matching and improve security against steganalysis. Even if the stego images arouse the attackers' suspicion, they cannot extract the secret message without the key.

5. Conclusions

In this research, a low-similarity CS-DCTR feature mapping approach is developed to increase database retrieval efficiency. The idea of coverless steganography guarantees both the inherent resistance of our system to steganalysis tools and the confidentiality of sensitive information. An attacker will never be able to read the secret message unless he obtains the key only known to the sender and receiver. Experiments show that the ideal value range of hiding bits of n per image in this scheme is $8, \cdots, 25$, considering both robustness and hiding capacity. However, the capacity of coverless systems, whether conventional or deep learning-based, is still significantly lower than that of embedding steganography. Therefore, more study is needed to address the capacity issue with coverless steganography.

6. Future Scope

With the development of security technology, complex steganography will provide more possibilities to serve various application scenarios. Database optimization is one of the primary issues, regardless. As is well known, for coverless steganography, the larger the number of hidden bits n, the more images are deserved in the database. If n is too large, it is almost impossible to collect enough images for the database. We may think of a way to make the length of the data hidden in each cover image variable rather than fixed to increase the message capacity based on a subsequence generation technique. The secret information segments can then adaptively match subsequences of any length.

Furthermore, geometrical attacks are another issue that must be overcome to improve the robustness of steganography systems. The problem of using an optimal distortion compensation mechanism to restore spatial synchronization for the data detector remains open. Future research should hopefully result in robustness against more varieties of visual distortions, including geometric transforms, contrast adjustments, and other lossy compressions.

Acknowledgments

This work was supported by Natural Science Foundation of Hunan Province (Grant No. 2019JJ50287, 2020JJ4248), Key scientific research projects of Hunan Provincial Department of Education (Grant No. 19A265, 21A0370), and China University Industry Research and Innovation Foundation-New Generation Information Technology Program (Grant No. 2020ITA09005).

References

[1] YANG, C. H., WENG, C. Y., WANG, S. J., et al. Adaptive data hiding in edge areas of images with spatial LSB domain systems. *IEEE Transactions on Information Forensics and Security*, 2008, vol. 3, no. 3, p. 488–497. DOI: 10.1109/TIFS.2008.926097

[2] MCKEON, R. T. Strange Fourier steganography in movies. In *2007 IEEE International Conference on Electro/Information Technology*. Chicago (IL, USA), 2007, p. 178–182. DOI: 10.1109/EIT.2007.4374540

[3] VALANDAR, M. Y., AYUBI, P., BARANI, M. J. A new transform domain steganography based on modified logistic chaotic map for color images. *Journal of Information Security and Applications*, 2017, vol. 34, no. 2, p. 142–151. DOI: 10.1016/j.jisa.2017.04.004

[4] COX, I. J., KILIAN, J., LEIGHTON, T., et al. Secure spread spectrum watermarking for images, audio and video. In *Proceedings of 3rd IEEE international conference on image processing*. Lausanne (Switzerland), 1996, vol. 3, p. 243–246. DOI: 10.1109/ICIP.1996.560429

[5] LIN, W. H., HORNG, S. J., KAO, T. W., et al. An efficient watermarking method based on significant difference of wavelet coefficient quantization. *IEEE Transactions on Multimedia*, 2008, vol. 10, no. 5, p. 746–757. DOI: 10.1109/TMM.2008.922795

[6] JANA, B. High payload reversible data hiding scheme using weighted matrix. *Optik - International Journal for Light and Electron Optics*, 2016, vol. 127, no. 6, p. 3347–3358. DOI: 10.1016/j.ijleo.2015.12.055

[7] MUKHERJEE, S., JANA, B. A novel method for high capacity reversible data hiding scheme using difference expansion. *International Journal of Natural Computing Research*, 2019, vol. 8, no. 4, p. 1–27. DOI: 10.4018/IJNCR.2019100102

[8] KODOVSKY, J., FRIDRICH, J., HOLUB, V. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 2011, vol. 7, no. 2, p. 432–444. DOI: 10.1109/TIFS.2011.2175919

[9] HOLUB, V., FRIDRICH, J. Phase-aware projection model for steganalysis of jpeg images. *Media Watermarking, Security, and Forensics*, 2015, vol. 9409, p. 259–269. DOI: 10.1117/12.2075239

[10] FRIDRICH, J., KODOVSKY, J. Rich models for steganalysis of digital images. *IEEE Transactions on information Forensics and Security*, 2012, vol. 7, no. 3, p. 868–882. DOI: 10.1109/TIFS.2012.2190402

[11] XU, G., WU, H. Z., SHI, Y. Q. Structural design of convolutional neural networks for steganalysis. *IEEE Signal Processing Letters*, 2016, vol. 23, no. 5, p. 708–712. DOI: 10.1109/LSP.2016.2548421

[12] BOROUMAND, M., CHEN, M., FRIDRICH, J. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 2018, vol. 14, no. 5, p. 1181–1193. DOI: 10.1109/TIFS.2018.2871749

[13] ZHOU, Z., SUN, H., HARIT, R., et al. Coverless image steganography without embedding. In *International Conference on Cloud Computing and Security (ICCCS)*. Nanjing (China), 2015, p. 123–132. DOI: 10.1007/978-3-319-27051-7_11

[14] CAO, Y., ZHOU, Z. L., SUN, X. M. Coverless information hiding based on bag-of-words model of image. *Journal of Applied Sciences*, 2016, vol. 34, no. 5, p. 527–536. DOI: 10.3970/cmc.2018.054.197

[15] ZHENG, S., LIANG, W., LING, B., et al. Coverless information hiding based on robust image hashing. In *International Conference on Intelligent Computing (ICICCS)*. Madurai (India), 2017, p. 536–547. DOI: 10.1007/978-3-319-63315-2_47

[16] LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, vol. 60, p. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94

[17] CAO, Y., ZHOU, Z. L., SUN, X. M., et al. Coverless information hiding based on the molecular structure images of material. *Computers, Materials & Continua*, 2018, vol. 54, no. 2, p. 197–207. DOI: 10.3970/cmc.2018.054.197

[18] ZHANG, X., PENG, F., LONG, M. Robust coverless image steganography based on DCT and LDA topic classification. *IEEE Transactions on Multimedia*, 2018, vol. 20, no. 12, p. 3223–3238. DOI: 10.1109/TMM.2018.2838334

[19] LIU, Q., XIANG, X., QIN, J., et al. Coverless steganography based on image retrieval of densenet features and DWT sequence mapping. *Knowledge-Based Systems*, 2019, vol. 192, p. 1–15. DOI: 10.1016/j.knosys.2019.105375

[20] LUO, Y., QIN, J., XIANG, X., et al. Coverless image steganography based on multi-object recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020, vol. 31, no. 7, p. 2779–2791. DOI: 10.1109/TCSVT.2020.3033945

[21] ZHOU, Z., CAO, Y., WANG, M. Faster-RCNN based robust coverless information hiding system in cloud environment. *IEEE Access*, 2019, vol. 7, p. 1–7. DOI: 10.1109/ACCESS.2019.2955990

[22] LUO, Y. J., QIN, J., XIANG, X., et al. Coverless image steganography based on image segmentation. *Computers, Materials and Continua*, 2020, vol. 64, no. 2, p. 1281–1295. DOI: 10.32604/cmc.2020.010867

[23] LIU, Q., XIANG, X., QIN, J., et al. A robust coverless steganography scheme using camouflage image. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, vol. 32, no. 6, p. 1–14. DOI: 10.1109/TCSVT.2021.3108772

[24] LIU, M. M., ZHANG, M. Q., LIU, J., et al. Coverless information hiding based on generative adversarial networks. *arXiv*, 2017, p. 1–14. DOI: 10.48550/arXiv.1712.06951

[25] HU, D., WANG, L., JIANG, W., et al. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, 2018, vol. 6, p. 38303–38314. DOI: 10.1109/ACCESS.2018.2852771

[26] DI, F., LIU, J., ZHANG, Z., et al. Somewhat reversible data hiding by image to image translation. *arXiv*, 2019, p. 1–16. DOI: 10.48550/arXiv.1905.02872

[27] CHEN, X., ZHANG, Z., QIU, A., et al. A novel coverless steganography method based on image selection and stargan. *IEEE Transactions on Network Science and Engineering*, 2020, vol. 9, no. 1, p. 1–12. DOI: 10.1109/TNSE.2020.3041529

[28] CHOI, Y., CHOI, M., KIM, M., et al. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City (UT, USA), 2018, p. 8789–8797. DOI: 10.1109/CVPR.2018.00916

[29] HOLUB, V., FRIDRICH, J. Low-complexity features for jpeg steganalysis using undecimated DCT. *IEEE Transactions on Information forensics and security*, 2014, vol. 10, no. 2, p. 219–228. DOI: 10.1109/TIFS.2014.2364918

[30] MA, S., ZHAO, X. Steganalytic feature based adversarial embedding for adaptive jpeg steganography. *Journal of Visual Communication and Image Representation*, 2021, vol. 76, no. 3, p. 1–12. DOI: 10.1016/j.jvcir.2021.103066

[31] SEDIGHI, V., FRIDRICH, J. Histogram layer, moving convolutional neural networks towards feature-based steganalysis. *Electronic Imaging*, 2017, vol. 2017, no. 7, p. 50–55. DOI: 10.2352/ISSN.2470-1173.2017.7.MWSF-325

About the Authors . . .

Lina TAN received the B.S. degree in Computer Science and the M.S. degree in Computer Software and Theory from Xiangtan University, Hunan, China, in 2003 and 2006, respectively, and the Ph.D. degree in Computer Application Technology from Hunan University, Hunan, China, in 2012. Her research interests include information security, digital watermarking, image processing, and pattern recognition.

Jiajun LIU (corresponding author: ljj20191005@163.com) was born in Hunan, China in 1999, and received the B.S. degree in Computer Science and Technology from Hunan University of Technology and Business in 2021. Currently, he is pursuing a M.S. degree in Computer Technology at the same university. His research interests include image steganography and network security.

Yu ZHOU was born in 1997. He received his M.S. degree from Hunan University of Technology and Business in 2023. His main research is image steganography.

Rongyuan CHEN received the B.S. degree in Computer Science from Nanjing University, Jiangsu, China, in 2002, the M.S. degree in Computer Software and Theory from Changsha University of Science & Technology, Hunan, China, in 2005, and the Ph.D. degree in Photogrammetry and Remote Sensing from Wuhan University, Hubei, China, in 2010. He is currently a Professor at Hunan University of Technology and Business, Hunan, China. His research interests include image processing and data mining.