Hindawi
Security and Communication Networks
Volume 2024, Article ID 5034640, 22 pages
https://doi.org/10.1155/2024/5034640

Research Article

A Robust Coverless Image Steganography Algorithm Based on Image Retrieval with SURF Features

Fan Li, Chenyang Liu, Zhenbo Dong, Zhibo Sun, and Weipeng Qian

School of Information and Communication Engineering, Hainan University, Haikou 570228, China

Correspondence should be addressed to Chenyang Liu; chenyang9015@163.com

Received 3 January 2024; Revised 22 April 2024; Accepted 3 May 2024; Published 18 May 2024

Academic Editor: Zahid Mehmood

With the advancement of image steganography, coverless image steganography has gained widespread attention due to its ability to hide information without modifying the carrier of images. However, existing coverless image steganography methods often require both communicating parties to transmit an amount of additional information including image blocks' locations or a large number of parameters, which will raise a serious suspicion. In light of this issue, we propose a robust coverless image steganography algorithm based on Speeded-Up Robust Features (SURF). Firstly, the proposed method allows both communicating parties to independently create multiple coverless image datasets (CIDs) using random seeds. Then, a mapping rule is designed for creating one-to-one correspondence between hash sequences and images in CIDs. Finally, the secret information will be carried by the images whose hash sequences are equal to the secret segments. At the receiver side, the robust SURF of images is utilized to retrieve the secret information. Experimental results demonstrate that the proposed algorithm outperforms other methods in terms of capacity, robustness, and security. Furthermore, it is worth noting that the proposed method eliminates the need to transmit a large amount of additional information, which is a significant security issue in existing coverless image steganography algorithms.

1. Introduction

With the widespread use of digital images and the popularity of social platforms, people often need to transmit and share images in their daily lives. In certain specific situations, users may wish to transmit sensitive information in a covert manner to protect personal privacy or engage in secret communication. Image steganography refers to the transmission of secret information by embedding it within digital images, making it visually imperceptible. In the context of secure communication, image steganography can be used for covert message delivery, allowing communication parties to hide sensitive content within seemingly ordinary images. It can also be used for copyright protection by embedding digital watermarks or copyright information to track and verify the origin and usage of an image.

The process of traditional image steganography techniques [1–7] typically involves embedding secret information by dispersing it among the pixel values or specific areas of an image and utilizing the redundancy of the image and the perceptual limitations of the human visual system to conceal the information. To extract the secret information, the corresponding key and extraction algorithm are required. Many outstanding image steganography algorithms have been developed [8–10]. Subramanian et al. [11] proposed the use of an end-to-end convolutional neural network to hide secret information, achieving a significant capacity. Lan et al. [12] designed a reversible network to hide information based on frequency coefficients, which demonstrated excellent resistance against JPEG compression attacks. Yang et al. [13] use a generative adversarial network (GAN) to hide secret messages in images. The sender maps the message to latent vectors, which are then used to generate a stego image that appears normal. The receiver can extract the message by optimizing the recovered latent vectors. The method is robust and has been tested with different GAN architectures, noise levels, and datasets. However, traditional image steganography not only

degrades the visual quality of the carrier image but also poses security concerns as it can be detected by steganalysis algorithms [14–16]. To address this issue, the concept of coverless steganography algorithms has emerged. These algorithms do not require modifying the carrier image itself and have the ability to completely evade detection by steganalysis algorithms. Specifically, coverless steganography algorithms connect secret information with the carrier through designed mapping rules. Coverless steganography does not mean that no carrier is required to transmit secret information. Rather, it means that no modifications are made to the carrier itself.

For coverless image steganography, the sender converts the secret message into binary bits and selects images to represent them according to the designed mapping rules. The receiver, following the same mapping rules, retrieves the binary bits and converts them back into the original secret message. Since no modifications are made to the image itself, it becomes harder for observers or unauthorized individuals to detect the hidden information. This makes coverless image steganography applicable in covert communication scenarios of real life, such as intelligence dissemination, secure communication, and covert exchanges.

The existing coverless steganographic algorithms for establishing the relationship between images and hash sequences can be divided into two types: generative-based and mapping-based algorithms. As the pioneer of generative-based methods, Zhou et al. [17] proposed generating hash sequences by averaging pixel blocks of scanned subimages. Subsequently, Zhang et al. [18] proposed a more robust approach to generate hash sequences by scanning the DCT coefficients of subimages. Then, Liu et al. [19] utilized the DWT decomposition of images and scanned the low-frequency components to generate hash sequences. In a study conducted by Zheng et al. [20], Scale-Invariant Feature Transform (SIFT) features were utilized to create a hash sequence. Building upon these works, Yuan et al. [21] presented their approach, which involved combining SIFT features and bagged features. To enhance resilience against geometric attacks, Luo et al. [22] introduced a faster region-based convolutional neural network (faster-RCNN). In the context of system security, Liu et al. [23] proposed an improvement by transmitting disguised images instead of steganographic images to the receiver. Although generative-based coverless steganography developed a lot, some common defects exist in it as follows. To cover all the secret information, generative-based methods often require a large image database. Firstly, more storage space is needed, and even using cloud storage will increase related costs. Secondly, managing and maintaining a large image database becomes more complex, and the quality of the data can impact the accuracy and reliability of steganographic algorithms. Typically, it is necessary to select specific regions of an image to generate the hash sequence for generative-based methods, which not only requires additional transmission of supplementary information but also raises security concerns. Also, these pieces of information need to be transmitted along with the image as additional information. Firstly, there is a transmission cost, and secondly, they are easily detected by surveillance personnel for tampering and deletion, leading to errors in extracting secret information. The most significant issue is that even slight changes in the image features used to generate the hash sequence can render the recipient unable to extract the secret information accurately, especially when the image is subjected to attacks. Therefore, there is a high demand for robustness in image features to ensure their resilience against various disturbances and attacks. In addition, in generative-based algorithms, it is typically required that the number of image features must be equal to the number of secret information, which results in lower capacity for generative-based algorithms.

Considering the issues related to security, robustness, and capacity in generative-based coverless image algorithms, recently, another type of coverless algorithm has emerged [24–31]. We refer to this as mapping-based coverless algorithms. Deep cross-modal hashing-convolutional neural network (DCMH-CNN) [32], proposed by Zou et al., is the most prominent example. The convolutional neural network (CNN) used in the DCMH-CNN is derived from the work of Jiang et al. as presented in DCMH [33]. A CNN was employed to extract high-dimensional image features, which were then used as deep hashes for image representation. Then, the K-means unsupervised learning algorithm was employed to cluster and build a small coverless image dataset (CID) using the high-dimensional image features. Finally, a mapping table was established to match the images in the CID with the hash sequences. Upon receiving an image, the receiver used the same CNN to extract high-dimensional features. Following the same mapping rule, the receiver identified the corresponding secret information. Compared to generative-based algorithms that require a large image database, mapping-based algorithms significantly improve the utilization of the original image database by constructing a much smaller CID. Moreover, mapping-based algorithms overcome the security concern of generative-based algorithms by not requiring additional transmission of supplementary information, such as block information. Furthermore, the utilization of the K-means algorithm in the DCMH-CNN [32] allows the number of image features no longer being required to be equal to the number of secret information segments. However, the DCMH-CNN [32] algorithm also has certain limitations and challenges. In terms of security, the DCMH-CNN [32] algorithm overlooks the repeated occurrence of secret segments. This results in the DCMH-CNN [32] algorithm requiring multiple repetitions of the same image to transmit a single repeated secret segment, which can easily raise suspicion from observers. In addition, both the sender and receiver need to use the same network for feature extraction and clustering, which requires the transmission of a considerable amount of network information between them. In addition, in terms of robustness, the accuracy of secret information extraction at the receiver's end can be further improved, especially the robustness against attacks such as central cropping, translation, speckle noise, and severe rotation.

Based on the analysis above, it can be concluded that the recently emerged mapping-based coverless image steganography has several evident advantages compared to earlier generative-based algorithms. However, they also face unresolved security and robustness issues. In light of this, we propose a strong robust mapping-based coverless algorithm.

The main process of the proposed algorithm involves the following steps: Firstly, select a publicly available image database. Then, use a random seed to generate random number sequences. When the values of the random seeds are fixed, the generated random sequences are the same, after that create multiple subimage databases by using the random number sequences. Each established subimage dataset contains same number of images, and these images within the subimage database will be used for transmitting secret information. We refer to each subimage database as CID. Subsequently, each image within the CID is matched with a binary hash sequence through a designed mapping rule. The sender divides the secret information into segments with the same length as the hash sequences. Based on the content of each secret information segment, the sender selects the corresponding image from the CID as the carrier for transmitting the secret information. The receiver first establishes multiple CID datasets consistent with the sender using the same random seed. After receiving the images, the robust feature Speeded-Up Robust Features (SURF) is extracted from the images for retrieving the original images from the CID. The reason we adopted SURF is because it possesses the following characteristics compared to other features: scale invariance, rotation invariance, robust feature descriptors, and fast feature computation depth. Finally, the corresponding secret information can be extracted using the same mapping rule. The main contributions of the paper are as follows:

(1) We address several important security issues existed in both generative-based and mapping-based coverless image steganography algorithms. Firstly, we utilizes a mapping approach to establish a one-to-one correspondence between CID images and hash sequences, effectively avoiding the drawback of generative-based algorithms that require transmitting a large amount of additional information. Secondly, we generate multiple CIDs, resolving the security concern of the DCMH-CNN algorithm, which relies on a single CID and may require repetitive transmission of the same image. Thirdly, the use of a random seed for image selection in constructing the CID significantly eliminates the need for transmitting a large number of parameters of a feature extraction network, which also reduces the communication risk between two parties.

(2) The proposed algorithm exhibits stronger robustness against various attacks on images. We break the symmetric structure in the DCMH-CNN algorithm, where both the sender and receiver need to extract consistent features, and the used features requires high robustness against various attacks. We only utilize the SURF of images for original image retrieval at the receiver's end. The SURF, introduced by Herbert Bay et al. [34] in 2006, and known for its fast speed and strong robustness is commonly used for feature extraction and matching in images. As a result, the requirement for robust feature design and extraction is significantly reduced.

2. Related Work

In this section, the basic principles of the coverless image steganography algorithm and the characteristics of generative-based and mapping-based algorithms are first discussed. Then, a comparison is made among existing coverless image steganography algorithms in terms of capacity, robustness against attacks, and security. Finally, we compare our proposed method with existing algorithms to identify the problems it solves.

2.1. Characteristics of Generative-Based and Mapping-Based Steganography. Coverless image steganography refers to a technique where the original carrier image remains unaltered when transmitting secret information, and instead, a mapping rule is established between the carrier image and the secret information, where each carrier image corresponds to one or more hash values. The transmission of secret information is achieved by selecting the corresponding carrier image whose hash value is equal to the secret information. For example, if the secret information is "101..11," the corresponding carrier image represents "101..11" is selected and transmitted. In 2015, Zhou et al. [17] first proposed the concept of coverless image steganography, whose hash values were generated by the average pixel values of scanned image blocks. Subsequently, researchers developed different methods to improve the robustness against various image attacks by using various features to generate hash values. These methods, which rely on image characteristics to generate hash values, are referred to as generative-based coverless image steganography algorithms, such as those proposed by Zhang et al. [18], Liu et al. [19], Zheng et al. [20], Yuan et al. [21], Luo et al. [22], and Liu et al. [23]. As shown in Figure 1, it is a simple generative-based coverless image steganography method. The image is first divided into 3×3 blocks. The average pixel value of each block is calculated, and the hash sequence is obtained by zigzag scanning each block. In Figure 1, if the average pixel value of the first scanned block is smaller than that of the second block, it is represented as "0." If the average pixel value of the third block is larger than that of the fourth block, it is represented as "1." This process is repeated for all blocks, resulting in the hash sequence "00101010" corresponding to the image. Thus, we can see that the hash sequences are generated directly based on the image pixels, and that is why we call these kinds of coverless steganography algorithms the generative-based methods.

Another approach is the mapping-based coverless image steganography algorithm, and unlike generative-based algorithms, a mapping rule is manually designed for mapping-based algorithms, which is used to hash values ranging from

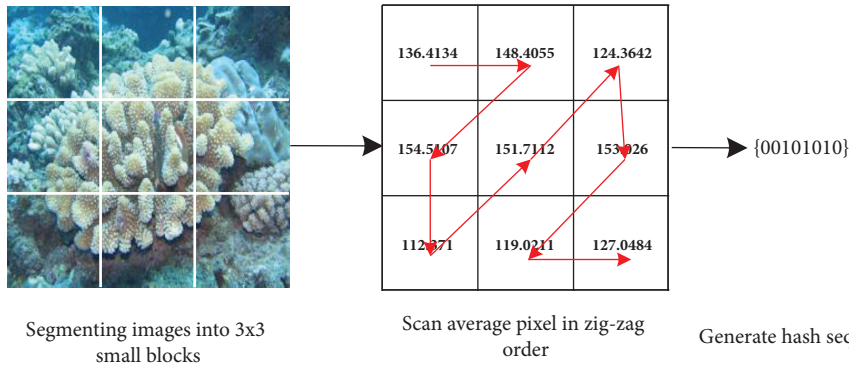

Figure 1: Generative-based coverless image steganography.

000...00 to 111...11. For instance, Zou et al. [32] and Luo et al. [35] used neural networks to extract high-dimensional features from images and sorted them based on these features to establish a mapping relationship between images and hash sequences. As shown in Figure 2, it is a mapping-based coverless image steganography method. First, features are extracted from each image. Then, based on the extracted features, the images are sorted according to rules set by individuals, such as feature value magnitude or feature value average. Finally, the hash sequence is assigned to each image in the order determined by the sorting, resulting in the completion of the hash sequence corresponding to each image. In Figure 2, for example, K images are chosen from the original dataset, after sorting in some way, the hypothetical 10th image is ranked first, then its corresponding hash sequence will be mapped to be "0000...00." The 6th image is ranked second, corresponding to "0000...01." This process continues until the 39th image, which corresponds to "1111...11." So, we can see that the mapping-based method is totally different from the generative-based method, and it is usually more robust because its hash sequences are mapped to the images according to some kinds of features instead of generated from the image blocks.

Until now, generative-based and mapping-based methods are the main popular coverless image steganography. Table 1 provides a comparison of these two types of coverless image steganography algorithms. From the comparison in Table 1, we can see that both types of coverless image steganography algorithms establish a one-to-one correspondence between images and hash sequences to achieve the steganographic effect. Generative-based coverless image steganography algorithms rely on images to generate hash sequences, and it is possible for multiple images in an image database to produce the same hash sequence. In order to cover all hash values from 000...00 to 111...11, generative algorithms usually need a large image database to search for suitable carrier images. By contrast, mapping-based coverless image steganography algorithms require a smaller image database because mapping rules are manually designed to match images with hash values. In the following analysis, we will summarize the existing coverless image steganography algorithms from three important and typical aspects: capacity, robustness, and security.

2.2. Performance and Analysis. In Table 2, we analyze four existing algorithms, among which LDA_DCT [18] and DenseNet_DWT [19] are two generative-based coverless image steganography algorithms that achieved good results. Faster-RCNN [35] and DCMH-CNN [32] are recently proposed mapping-based coverless image steganography algorithms.

Based on the comparison in Table 2, we can observe that in terms of capacity, LDA_DCT [18] and DenseNet_DWT [19] performed worse than faster-RCNN [35] and DCMH-CNN [32]. This is because generative-based coverless image steganography algorithms required additional images to transmit positional information, whereas mapping-based coverless image steganography algorithms only needed both communicating parties to know the established mapping rules. As a result, generative-based algorithms generally exhibit lower capacity compared to mapping-based algorithms.

Robustness affects the accurate extraction of the secret information and is an important concern for researchers. In terms of robustness against various image attacks, LDA_DCT [18] and DenseNet_DWT [19] performed worse than faster-RCNN [35] and DCMH-CNN [32]. This is because the hash sequences in generative-based algorithms highly rely on image pixels, and even a slight disturbance in the image can cause inconsistencies between the generated hash sequence and the sender's sequence. On the other hand, mapping-based algorithms, by establishing individual CID image database for both parties, only require the receiver to find the corresponding image in the CID image database and extract the corresponding hash sequence based on the established mapping rules. This significantly enhances the robustness. The impact on robustness also depends on the specific algorithm. For example, among the mapping-based algorithms, faster-RCNN [35] exhibited lower robustness than the recently proposed DCMH-CNN [32]. This is

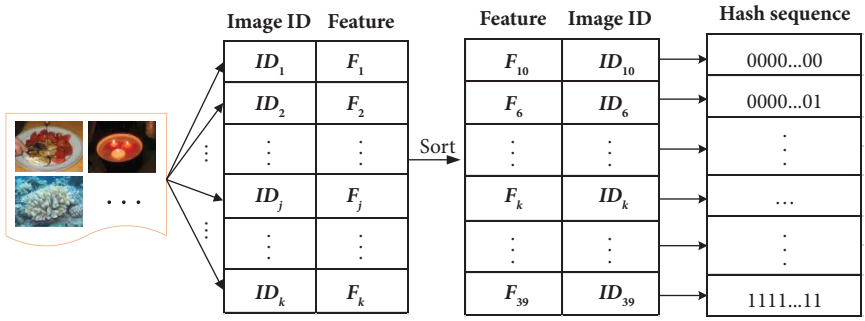

Figure 2: Mapping-based coverless image steganography.

because the DCMH-CNN [32] constructed the CID using images with the largest feature differences. Furthermore, it is worth noting that existing coverless image steganography algorithms require both parties to use the same features, which demand strong feature robustness.

Security is a crucial aspect that researchers focus on since algorithms can only be applied in real-life scenarios with sufficient security guarantees. LDA_DCT [18] and DenseNet_DWT [19] required the transmission of image positional information for generating the hash sequence. This is a common issue in generative-based coverless image steganography algorithms. Transmitting the position information of image blocks poses security risks due to the large amount of data that can be intercepted and tampered with by eavesdroppers. Similarly, the faster-RCNN [35] and DCMH-CNN [32] required both parties to share the same feature extraction network including the exactly same parameters, which involved transmitting a large amount of data that can also be intercepted and tampered with, leading to security concerns. In addition, all these four methods faced the security issue of repeatedly transmitting the same image. For example, if the secret information is 1 MB and each image can hide 8 bits of capacity, each image needs to be repeatedly transmitted four times on average to distribute the fragments of the secret information. This is assuming the best-case scenario. Repeated transmission of images can easily raise suspicion and lead to interception and deletion of the secret information.

Based on the analysis above, it can be concluded that the recently emerged mapping-based coverless image steganography have several evident advantages compared to earlier generative-based algorithms. However, they also face unresolved security and robustness issues. In light of this, we adopt the basic mapping-based coverless steganography method and focus on solving the security problems that the existing mapping-based methods have and improve the robustness further. Several crucial design improvements including more robust image retrieval features, breaking the symmetric structure of the DCMH-CNN [32] algorithm, randomly constructing multi-CID image databases instead of only one CID in previous methods, manually mapping the hash sequences to one whole image instead of generating hash sequences based on image blocks will be designed in this paper. These improvements would address several significant security issues in both mapping-based and generative image algorithms, while also achieving high capacity and enhancing the robustness to image attacks as we mentioned in Table 2.

3. The Proposed Coverless Steganography Algorithm Based on SURF

3.1. Overview. The overall framework of the algorithm, as shown in Figure 3, consists of the following components: prepoceessing the original image dataset, constructing multi-sub-CIDs, establishing a mapping rule of hash sequences, embedding secret information, the image retrieval by using SURF, and secret information extraction.

In Figure 3, the sender is represented above the dashed line, and the receiver is below it. Firstly, select a publicly available graphics dataset, represented as "original image dataset" in Figure 3. Preprocessing is applied to generate the "sorted image dataset". In the sorted image dataset, the images are arranged in ascending order based on their average pixel values and named accordingly. The specific operations are detailed in Section 3.2.

Then, a random array such as "1610, 155, 861, . . ., 172, . . ., 1812" as shown in Figure 3 is generated by using a random seed. Then, the images whose sequence numbers are the same as the random array such as 1610, 155, 861, . . ., 172, . . ., 1812 will be selected from the sorted image dataset to form the first CID, after that, the chosen images are removed from the sorted image dataset, then the second CID can be constructed in the same way with the same array in the left sorted image dataset. This process is repeated to generate multiple CIDs, as explained in detail in Section 3.3.

Following that, a corresponding mapping rule is established, such that each image in the sub-CIDs corresponds to a binary hash sequence from 000. . .000 to 111. . .111. The specific operations are described in Section 3.4.

To carry the secret information using the images in CIDs, the sender divides the secret message into segments and selects the images whose mapping hash sequences are equal to the message segments as cover images. This process is explained in detail in Section 3.5.

Lastly, at the receiver's end, the receiver first extracts the robust SURF (Speeded-Up Robust Features) for all the received images, which might be destroyed by various attacks, and then uses the nearest distance method to retrieve the

Table 1: Comparison of the two types of coverless image steganography algorithms.

Algorithm	Is there a one-to-one correspondence between images and hash values?	Rely on generating a hash sequence based on the image?	Necessary to manually establish a mapping rule?	Need a large image database?
Generative-based coverless image steganography	Yes	Yes	No	Yes
Mapping-based coverless image steganography	Yes	No	Yes	No

Table 2: The performance of existing coverless image steganography algorithms in terms of capacity, robustness, and security.

Algorithm	Capacity	Robustness to various image attacks	Security problem
LDA_DCT [18]	Lower	Lower	Requires transmitting the image blocks' positions, and repeatedly transmitting the same image
DenseNet_DWT [19]	Lower	Lower	Requires transmitting the image blocks' positions, and repeatedly transmitting the same image
Faster-RCNN [35]	High	Moderate	Requires sharing the same feature extracting networks, and repeatedly transmitting the same image
DCMH-CNN [32]	High	High	Requires sharing the same feature extracting networks, cluster center K, and repeatedly transmitting the same image
The proposed method	High	High	No any additional information No repeatedly transmitting the same image

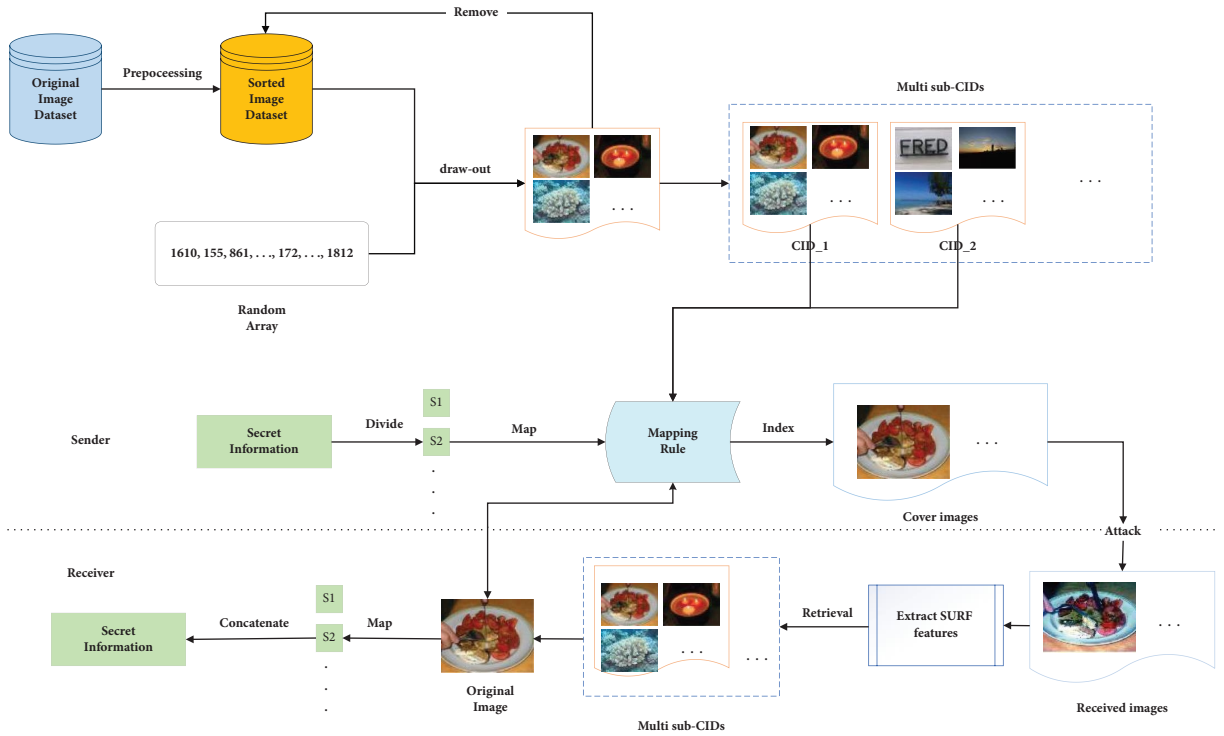

Figure 3: Systematic overview of robust coverless image steganography algorithm employing image retrieval with SURF features.

original image for each received image. Finally, with the predefined mapping rule, the content of the secret segment corresponding to the retrieved original image can be identified and concatenated to reconstruct the complete secret message. The specific operations are explained in detail in Section 3.6.

3.2. Preprocessing the Original Image Dataset. Any public image database which contains enough images can be used as the original database for constructing CIDs, such as the popular datasets: ImageNet, COCO, Open Images, Flickr, and SUN. First, both communicating parties need to preprocess the original image dataset in the same way, that is to say, sort the original images by using the same image features, such as the magnitude of DC coefficients or low-frequency DWT coefficients. In this paper, the original images are sorted in ascending order of the average pixel value. The preprocessing process is shown in Figure 4.

Firstly, we calculate the average pixel value of each image and then sort the images in the original image dataset in ascending order of the average pixel value. For example, in Figure 4, there are M images in the original image dataset. After sorting, the image of ID_j which has the lowest average pixel value is ranked first, and then the image of ID_{96} which has the second lowest average pixel value will be ranked second. This process continues, sorting the images in ascending order based on their average pixel values until ID_{50} with the highest average pixel value is ranked last.

3.3. Constructing Multi-Sub-CIDs. Then, the images of each sub-CIDs can be selected from the sorted image dataset by using a random array with a fixed random seed. Figure 5 shows the process of constructing multi-sub-CIDs and mapping rules.

First, generate the random sequence using a random seed. The random sequence can be generated by the program "rng (seed)" in MATLAB or "random.seed (seed)" in Python, where "seed" is the seed value. The seed value typically represents an integer that serves as the starting point for the random number generator algorithm. It is important to note that the range of valid seed values may be limited in different programming languages. In MATLAB, the seed value must be a 32-bit signed integer, which means it should fall within the range of -2^{31}–$2^{31} - 1$. In Python, the seed value can be any integer, but it is internally converted to a 32-bit signed integer, limiting the range to -2^{31}–$2^{31} - 1$. For example, let us generate a random sequence with a length of 256. In this case, we choose the random seed value of 42. By using the program of `rng (42) and "random sequence = randi ([0, 500], 1, 256)" in MATLAB, a random sequence of length 256 with values ranging from 0 to 500 can be obtained. When the "seed" values of both communication parties are the same, the random array is the same. This ensures consistency and synchronization between the parties, facilitating reliable and effective communication. In addition, both communication parties also need to set a random sequence length M to constrain the number of images in each CID.

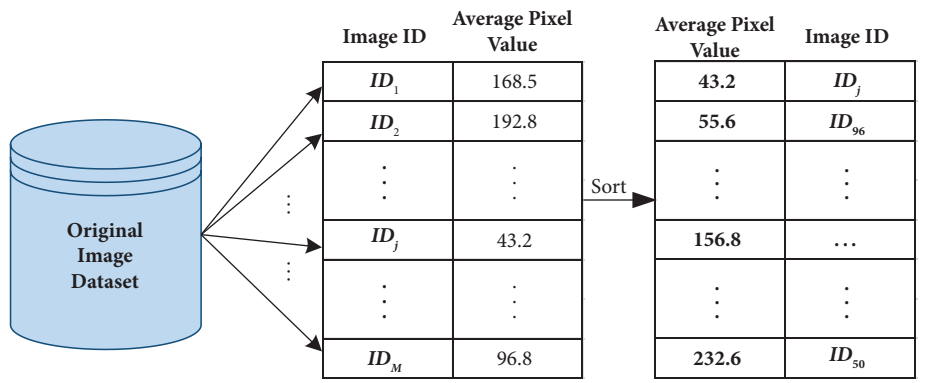

Figure 4: Preprocessing the image dataset.

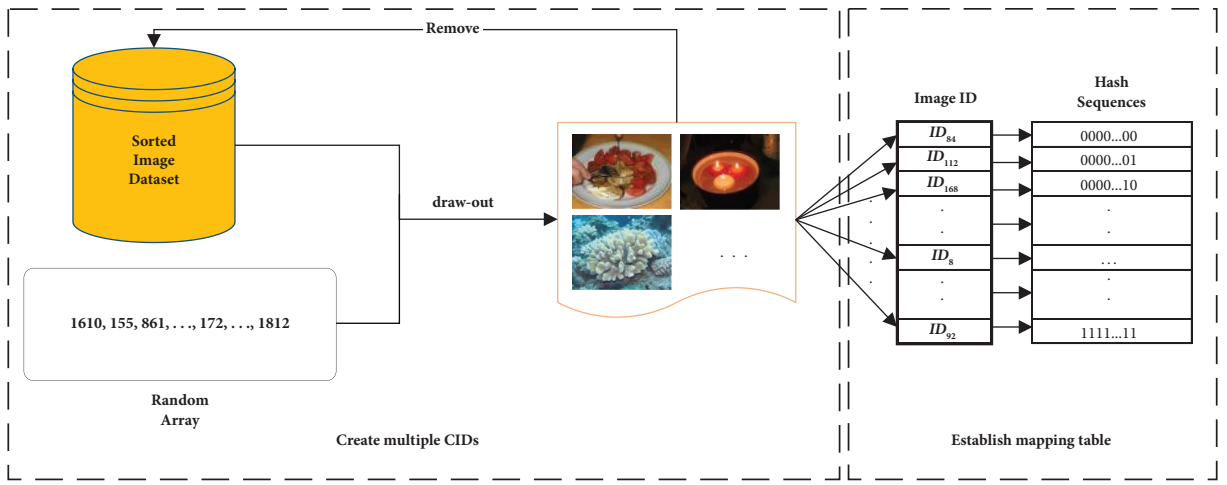

Figure 5: Process of constructing multi sub-CIDs and mapping rule.

Suppose that the random sequence generated by a given seed is 1610, 155, 861, . . ., 172, . . ., 1812, this means that the image with the average pixel value ranked 1610, 155, 861, . . ., 172, . . ., 1812 will be selected as the first CID. Therefore, the length of a random sequence equals to how many images are needed to build a CID subdataset. Suppose each CID contains M images, after constructing the first CID, the selected M images are removed from the original image dataset. The left images are sorted in the same way, and another M image can be chosen from it to form the second CID. This process is repeated N times to finally get N sub-CIDs. The parameter M is explained in Section 3.4, and the value of N is explained in Section 3.5.

3.4. Establishing the Mapping Rule of Hash Sequences. As we know, the images are selected from the original image database to form multiple CIDs based on the random sequence. At the same time, the mapping rule can be established following the random sequence. As shown in Figure 5, if the first number of the sequence is 1610, the 1610th image in the sorted original database will be selected as the first image of the CID, and then the 1610th image is stored in the first row of the mapping table, the hash sequence corresponding to it is given to be "0000. . .00". Then, following the order of the images in the CID, the corresponding hash sequence is added by one. This process continues until the last image in the CID is mapped, which corresponds to "1111. . .11."

The relationship between the number of images in a CID subdatabase M and the length of the hash sequence L can be represented as follows:

$$L = \log_2 M. \tag{1}$$

Please note that the length of the random sequence that needs to be generated is also M, and different lengths of random sequences can affect the length of the corresponding hash sequence for images. For example, if you want an image to correspond to an 8-bit-long hash sequence, then each CID subdataset requires 2^8 images, and the length of the random sequence is 256.

3.5. Secret Information Hiding. The hiding of secret information for the coverless steganography algorithm refers to the process of selecting corresponding cover images from multi-sub-CIDs based on the secret information.

The process of hiding secret information is as follows:

Step 1: The sender first divides the secret message whose length of binary representation is G, into segments. Each segment is of the same length as the hash sequence L. As a result, the secret message will be divided into n segments:

$$n = \begin{cases} G\%L, & \text{if } G\%L = 0, \\ \lfloor G\%L \rfloor + 1, & \text{Otherwise}, \end{cases} \tag{2}$$

where the symbol $\lfloor . \rfloor$ represents the floor function. When the length G of the secret message is not divisible by the hash sequence length L, the last segment of the secret message, which has a length less than L, is padded with zeros to make its length equal to L. The number of added zeros is also recorded.

Step 2: Generate multi-sub-CIDs according to Section 3.4. Here, a **Key** as shown in (3) is needed to share between sender and receiver, so that both parties can construct the same CIDs.

$$\mathbf{Key} = (N, M, \text{Seed}). \tag{3}$$

In (3), N represents the number of sub-CIDs to be established. The value of N depends on the number of occurrences of the most frequently repeated secret message segment in the entire secret message. For example, if the secret message "101..11" appears 20 times, and it is the segment who repeat the most times among all secret message segments, then we need to establish 20 CIDs. M represents the number of images contained in each sub-CID. *Seed* represents the random seed used by the communication parties to establish the same random sequence, ensuring that the order of images is fixed. It is worth noting that unlike other algorithms where the receiver needs to receive additional information while receiving images, the proposed algorithm only needs to receive the images and the simple key.

Step 3: Construct the mapping relationship between images and hash sequences in each CID subdatabase, following the rules explained in Section 3.2.

Step 4: Then, based on the content of each secret message segment S_i, where $i = 1, 2, \ldots, n$, the mapping table is used to find the cover image for each segment. The selection rule is as follows: the corresponding image in the CID whose hash sequence is the same as the secret segment S_i is selected as the carrier. We refer to this selected image as the i-th cover image SI_i. For simplicity, let us assume that there are N sub-CIDs, each containing M images, then the selection process can be described as follows:

$$SI_i = I_{c,j}, \quad \text{if } S_i = HI_{c,j}, \tag{4}$$

$$1 \le c \le N, 1 \le i \le n, 1 \le j \le M, \tag{5}$$

where $I_{c,j}$ represents the chosen j-th image in the c-th sub-CID, and $HI_{c,j}$ means the mapping hash sequence of $I_{c,j}$. Repeat the above steps until all the secret segments S_i and recorded zeros are represented by corresponding cover images SI_i. It is important to note that first select images from the first sub-CID database when a repeat secret segment occurs, the current sub-CID image database will be skipped, and the search for an image corresponding to the same hash sequence will continue in the next CID subimage database.

Step 5: Repeat Step 4 until the entire secret message is embedded. If the last segment of secret information is padded with zeros, the number of zeros is converted into a binary sequence, and then the carrier image $SI_{padding}$ for it can be found out by using the same way as other secret segments. Finally, we can get the whole stego image set **SI**:

$$\mathbf{SI} = \left\{\mathbf{SI}_1, \mathbf{SI}_2, \ldots, \mathbf{SI}_n, \mathbf{SI}_{\text{padding}}\right\}. \tag{6}$$

Step 6: The sender transmits the steganographic images to the receiver.

The pseudocode for the information-hiding process is shown in Algorithm 1.

3.6. Secret Information Extraction. For the coverless steganography algorithm, the extraction of secret information involves accurately identifying the hash sequence corresponding to the received image sent by the sender. Therefore, we firstly employ an image retrieval method to locate the original image corresponding to the received image that might have undergone malicious attacks. Then, we match the generated hash sequence with secret information by using the established mapping table. The specific process for the receiver to extract the secret information is as follows:

Step 1: The receiver adopts the same method as the sender to preprocess and sort the original image database. Then, multiple sub-CIDs are independently established by using the key.

Step 2: Upon receiving the stego images, the received images are then subjected to retrieve the original images from all the CIDs with the help of a popular image feature SURF [34]. SURFs are chosen for retrieval in this paper because of their efficiency and robustness. The general process of calculating SURF for one image is shown in Figure 6.

(1) Scale-space construction: The image is subjected to scale-space construction using the Difference of Gaussians (DoG) method.

(2) Key point detection: In scale-space images, the scale-invariant Hessian matrix is adopted to detect extrema as key points.

(3) Key point localization: The localization of key points is achieved by accurately determining the positions of extrema in the DoG scale-space images. Figure 7 shows the localized key points for an image.

Input: Secret information **S**, sorted image database **IMG**, Key = $(N, M, Seed)$
Output: Stego image $\mathbf{SI} = \{\mathbf{SI}_1, \mathbf{SI}_2, \ldots, \mathbf{SI}_n, \mathbf{SI}_{\text{padding}}\}$
(1) Divide **S** into segments: $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$
(2) Generate multiple sub-CIDs $\mathbf{I} = \{\mathbf{I}_{1,1}, \mathbf{I}_{1,2}, \ldots, \mathbf{I}_{\text{N,M}}\}$ using sorted image database **IMG** and **Key**
(3) for $k = 1$: $\mathbf{I}_{N,M}$ do
(4) Generate a hash sequence for each image based on the order of the sequence $\mathbf{HI}_{c,j}, 1 \le c \le N, 1 \le j \le M$
(5) end for
(6) Update image index database
(7) for $i = 1$: n do
(8) Search for $\mathbf{SI}_i = \mathbf{I}_{c,j}$, if $\mathbf{S}_i = \mathbf{HI}_{c,j}$
(9) When $\mathbf{I}_{c,j}$ has already been used $\mathbf{SI}_i = \mathbf{I}_{c+1,j}$, **if** $\mathbf{S_i} = \mathbf{HI}_{c+1,j}$
(10) end for
(11) Record the number of zero fillings and map it to the last image $\mathbf{SI}_{\text{padding}}$
(12) Get stego images set = $\{\mathbf{SI}_1, \mathbf{SI}_2, \ldots, \mathbf{SI}_n, \mathbf{SI}_{\text{padding}}\}$ and transmit them to the receiver
(13) end.

Algorithm 1: Secret Information Hiding.

(4) Orientation calculation: Within the neighborhood of each key point, the dominant orientation is calculated to describe the key points' rotational invariance.

(5) Feature description: Within the neighborhood of each key point, descriptors for local image patches are calculated, and they are called SURFs.

Step 3: Following Step 2, we can extract SURF for all the images in CIDs and the query image, and then we match the query image with all the images in CIDs. Figure 8 illustrates the matching of 100 SURF points. The left image shows the original image, while the right one shows the image after a 30° rotation attack.

After the SURF matching, the indices and corresponding distance values of the matching pairs can be obtained. The distance values are metrics used to measure the similarity between two images. A smaller distance value indicates a higher similarity between the two images, while a larger distance value indicates a greater dissimilarity. In this paper, we adopt the average distance to measure the similarity between two images, which involve summing the distance values of all the matching pairs and dividing it by the number of matching pairs. Then, we can get the distances between the query image and all the images in the CIDs. Finally, the distance values are sorted, and the image corresponding to the smallest distance is selected as the matching result, which is noted as the retrieved image $\mathbf{I}_{c,j}$.

Step 4: According to the previously set mapping rule, the hash sequence $HI_{c,j}$ of the retrieved image $I_{c,j}$ for the i-th received image can be found out, and then the secret information $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_i, \ldots, \mathbf{S}_n\}$ can be represented by

$$\mathbf{S}_i = \mathbf{HI}_{c,j}, \tag{7}$$

$$1 \le i \le n, 1 \le c \le N, 1 \le j \le M. \tag{8}$$

If there is zero-padding information, for the last received image $\mathbf{SI}_{\text{padding}}$, the SURF matching is also used to retrieve the corresponding zero-padding quantity, and the corresponding number of zeros is then removed from the last segment of the secret information. Then, the extracted secret segments are concatenated to obtain the complete secret information **S**.

The process of secret information extraction is illustrated in Algorithm 2.

4. Experimental Results

4.1. Experimental Setup. All experiments in this paper were conducted on a personal computer equipped with an AMD R5 5600X CPU@3.50GHz and 16 GB of memory. The image processing functions from the Image Processing Toolbox in MATLAB R2018a were used to implement various types of image attacks, including noise and cropping.

For the experimental datasets, we employed three commonly used public image databases: INRIA Holiday (https://lear.inrialpes.fr/%7Ejegou/data.php), ImageNet (https://image-net.org), and Caltech-256 (https://data.caltech.edu/records/nyy15-4j048). The description of each dataset and the selection of label classes are explained as follows:

(1) The INRIA Holiday dataset consists of 1491 images, including natural landscapes, urban scenery, buildings, and other types of tourist photographs. The images in this dataset exhibit rich visual features and diversity, covering various environments, lighting conditions, and shooting angles. Furthermore, the images have a relatively high resolution.

(2) The ImageNet dataset contains over 15 million images, covering more than one thousand categories, including animals, objects, plants, scenes, and so on. Each image is associated with a corresponding category label, which can be used for tasks such as image classification, object detection, and scene understanding.

Figure 6: The process of calculating SURF features.

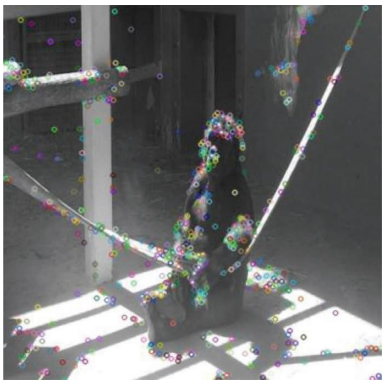

Figure 7: Key point localization.

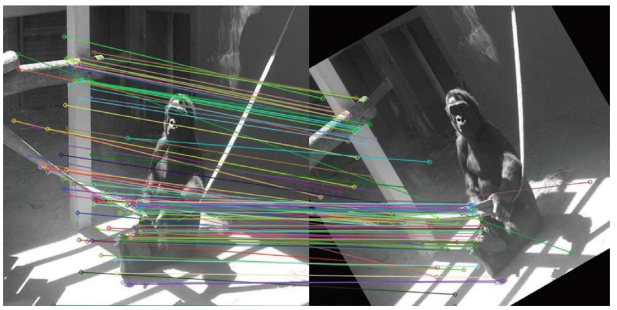

Figure 8: Feature point matching.

Input: Stego images $\mathbf{SI} = \{\mathbf{SI}_1, \mathbf{SI}_2, \ldots, \mathbf{SI}_n, \mathbf{SI}_{\text{padding}}\}$, sorted image database **IMG**, **Key** = (N, M, Seed)
Output: Secret information **S**
(1) Generate multiple sub-CIDs $\mathbf{I} = \{\mathbf{I}_{1,1}, \mathbf{I}_{1,2}, \ldots \mathbf{I}_{N,M}\}$, using sorted image database **IMG** and **Key**
(2) for $k = 1$: $\mathbf{I}_{N,M}$ do
(3) Generate a hash sequence for each image based on the order of the sequence $\mathbf{HI}_{c,j}, 1 \le c \le N, 1 \le j \le M$
(4) end for $i = 1$: n do
(5) Compute the SURF of the stego image $\mathbf{SI} = \{\mathbf{SI}_1, \mathbf{SI}_2, \ldots, \mathbf{SI}_n, \mathbf{SI}_{\text{padding}}\}$
(6) For each received image $\mathbf{I}_i$, match it with the most similar original image $\mathbf{I}_{c,j}$
(7) Find the hash sequence $\mathbf{HI}_{c,j}$ corresponding to the image $\mathbf{I}_{c,j}$
(8) $\mathbf{S}_i = \mathbf{HI}_{c,j}$
(9) get $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$
(10) end for
(11) Calculate the zero-padding information $\mathbf{SI}_{\text{padding}}$ (if it exists)
(12) Remove the number of trailing zeros from the last segment of the secret information, concatenate the secret information fragments, and obtain the complete secret information **S**
(13) end.

Algorithm 2: Secret Information Extraction.

(3) The Caltech-256 dataset was created by the California Institute of Technology (Caltech). It consists of 256 distinct object categories, and each category contains approximately 80 to 800 images. The images in this dataset are captured from real-world scenes and objects, covering a wide range of common objects, animals, plants, and everyday scenes. The Caltech-256 dataset is widely used for training and testing various computer vision algorithms.

For the INRIA Holiday dataset, all 1491 images were resized to a size of 512×512 for experimental purposes. In the ImageNet dataset, we selected two labels, "n014400764" (chicken) and "n01514668" (fish), totaling 2600 images. These images were resized to a size of 128×128 for experimentation. Regarding the Caltech-256 dataset, we chose seven labels: "001.ak47," "085.goat," "086.golden gate-bridge," "087.goldfish," "089.goose," "090.gorilla," and "092.grapes," with a total of 906 images. These images were resized to a size of 128×128 for experimentation.

These selection and resizing procedures aligned with the methods used in LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32], where LDA_DCT [18] and DenseNet_DWT [19] are the latest generative-based coverless image steganography methods, while DCMH-CNN [32] is the only mapping-based method. Therefore, we chose these three algorithms to compare with. In order to demonstrate the superiority of the proposed method, we reproduced the experiments of LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32] in the same datasets, rather than using the original data for a fair comparison.

4.2. Capacity. The capacity of existing coverless image steganography is usually measured by the number of images needed to transmit a fixed-length secret message. In the proposed algorithm, the capacity is determined by the number of images M in the CID. If the length of a secret message segment represented by one image is L, the number of images for constructing one CID is set to be

$$M = 2^L. \tag{9}$$

Each CID subdatabase contains M images as cover images; theoretically, as long as the original image database is large enough, the value of M can be sufficiently large too. This means that each image can represent a longer hash sequence, and thus the corresponding secret segment carried by each image is longer. As a result, fewer images are needed to send the whole secret message.

The number of images required to send a secret message of length G can be calculated as follows:

$$N_h = \left\lfloor \frac{G}{L} \right\rfloor, \tag{10}$$

where $\lfloor . \rfloor$ represents the ceiling function. However, selecting too many images can result in low retrieval efficiency for the receiver. For the sake of convenience, we select $L = 16$ to compare with other methods and calculate the number of images required to send secret messages with length of 1B, 10B, 100B, and 1 KB secret messages using various methods. Here, B represents byte, and 1B is equal to 8 bits. The comparative results are shown in Table 3.

The table lists the minimum number of cover images required to hide secret messages of different lengths under the given capacities. By comparing the results, it can be observed that the proposed method has a higher capacity compared to generative-based coverless algorithms: LDA_DCT [18] and DenseNet_DWT [19]. Compared to the mapping-based coverless algorithm DCMH-CNN [32], the proposed method has the same capacity.

4.3. Robustness. The communication between the two parties in transmitting secret images may encounter various geometric and nongeometric attacks. In this study, the same method as DCMH-CNN [32] is employed to simulate a series of attacks on the received images. The attack effects are shown in Figure 9, with the original image sourced from the Holidays dataset. All types of attacks and the corresponding parameters are presented in Table 4.

In the proposed method, the receiver retrieves the original image of the received one and applies the mapping rule to find the corresponding secret information. The accuracy of secret information extraction is defined as follows:

$$Acc = \frac{\sum_{i=1}^{n} f\left(EI_i\right)}{n} \times 100\%, f\left(EI_i\right) = \begin{cases} 1, & \text{if } EI_i = I_i, \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where I_i is the original i-th secret segment, and EI_i is the extracted i-th secret segment. As we know, the secret information is divided into n segments, so the accuracy is decided by how many the extracted secret segments are equal to the original ones. The defined accuracy is used to measure the robustness of the proposed method to various attacks, and the higher the accuracy is, the stronger the robustness becomes.

Tables 5–7 show the accuracy comparison results on the ImageNet, Holidays, and Caltech-256 datasets, respectively, and the average accuracy is listed in the last row of the tables. When comparing with other algorithms, we choose to have each CID database containing 256 images, which corresponds to a capacity of 8 bits.

The bold data in the tables represent the highest accuracy among the compared algorithms. The italicized data represent the second-highest accuracy. From the results, we can see that the proposed algorithm maintains accuracy above 90% for all attacks except for the 50% central cropping and color histogram equalization attacks. In most cases, the proposed algorithm achieves the highest or second highest accuracy under the same attack compared to other algorithms, and the average accuracy of the proposed algorithm is the highest in all three databases, at around 25% higher than the generative-based methods LDA_DCT [18] and DenseNet_DWT [19], and 4% higher than the mapping-based method DCMH-CNN [32]. Overall, the proposed algorithm demonstrates stronger robustness against different types of attacks.

Table 3: The comparison of the number of images required to transmit secret messages of different lengths and their respective capacities.

Algorithm	N_h				Capacity
	1B	10 B	100B	1 KB	
LDA_DCT [18]	2	7	55	548	15
DenseNet_DWT [19]	2	7	55	548	15
DCMH-CNN [32]	1	5	50	512	16
Our method	1	5	50	512	16

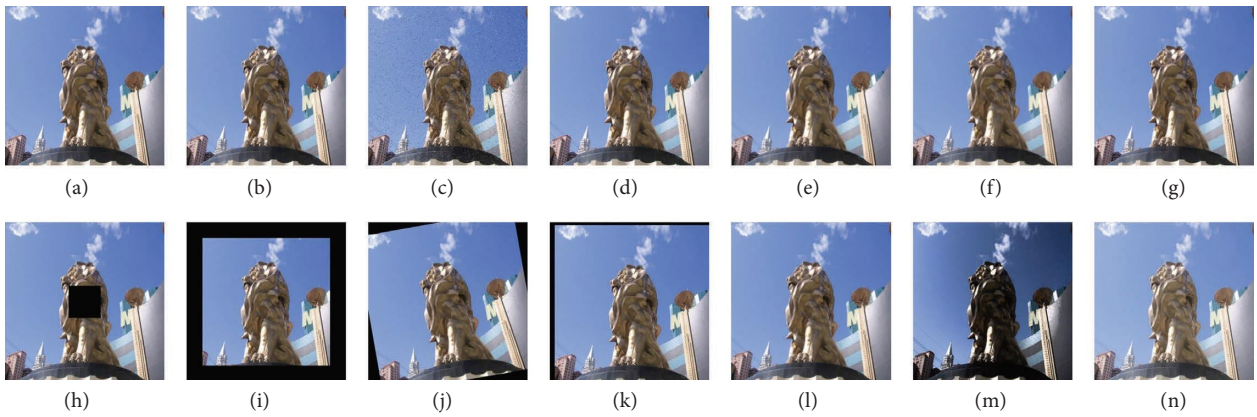

Figure 9: Some samples of attacked images in holidays: (a) JPEG compression: $Q = 90$, (b) Gauss noise: $\mu = 0$ and $\sigma = 0.001$, (c) salt and pepper noise: $\mu = 0$ and $\sigma = 0.001$, (d) Speckle noise: $\mu = 0$ and $\sigma = 0.01$, (e) Gauss low-pass filtering: 3×3, (f) mean filtering: 3×3, (g) median filtering: 3×3, (h) centered cropping: 20%, (i) edge cropping: 10%, (j) rotation: 10°, (k) translation: (16, 10), (l) scaling: 1.5, (m) color histogram equalization, and (n) gamma correction:0.8.

Table 4: Types of attacks performed on images in experiments and their parameters.

Image processing	The specific parameters
JPEG compression	The quality factors Q: 10%, 50%, and 90%
Gauss noise	The mean μ: 0, the variance σ: 0.001 and 0.005
Salt and pepper noise	The mean μ: 0, the variance σ: 0.001 and 0.005
Speckle noise	The mean μ: 0, the variance σ: 0.001 and 0.005
Gauss low-pass filtering	The window size: 3×3
Mean filtering	The window size: 3×3
Median filtering	The window size: 3×3
Centered cropping	Ratio: 20% and 50%
Edge cropping	Ratio: 10% and 20%
Rotation	Rotation angles: 10°, 30°, and 50°
Translation	In holidays: (80, 50), (160, 100), and (320, 200) In ImageNet, Caltech-256: (16, 10), (32, 20), and (40, 25)
Scaling	Ratio: 0.5, 0.75, 1.5, and 3
Color histogram equalization	None
Gamma correction	Factor: 0.8

Furthermore, to test the impact of different attack parameters on accuracy, we conducted a comprehensive comparison of the fluctuation range of accuracy for different intensity attacks across three image databases. For different intensity JPEG compression attacks, the LDA_DCT [18] algorithm exhibits a fluctuation range of 6.7%–8.2%, the DenseNet_DWT [19] algorithm has a fluctuation range of 2.0%–3.1%, and the DCMH-CNN [32] algorithm shows a fluctuation range of 2.3%–40.6%. In contrast, our algorithm has a much smaller fluctuation range of only 0.4%–1.2%. Regarding various types of noise attacks, the fluctuation range of the LDA_DCT [18] algorithm is 0%–5.8%, the DenseNet_DWT [19] algorithm has a fluctuation range of 0.4%–1.8%, and the DCMH-CNN [32] algorithm exhibits a fluctuation range of 0.4%–6.6%. In comparison, our algorithm has a significantly smaller fluctuation range of only 0%–0.4%. For rotation and translation attacks, we compared our

Table 5: Robustness comparison with LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32] in ImageNet dataset.

Processing	Size	LDA_DCT (%)	DenseNet_DWT (%)	DCMH-CNN (%)	Proposed (%)
JPEG	Q (10)	91.4	*98.0*	59.4	**98.8**
	Q (50)	*98.4*	98.0	94.1	**100.0**
	Q (90)	*99.6*	**100.0**	**100.0**	**100.0**
Gauss-N	σ (0.001)	94.1	98.8	*99.6*	**100.0**
	σ (0.005)	94.1	*98.0*	93.0	**100.0**
S&P-N	σ (0.001)	*98.4*	**100.0**	**100.0**	**100.0**
	σ (0.005)	94.1	98.8	*99.6*	**100.0**
Speckle-N	σ (0.01)	**99.6**	*99.2*	**99.6**	**99.6**
	σ (0.05)	89.8	*91.0*	73.8	**98.4**
Gauss-F	(3 × 3)	*99.2*	**100.0**	**100.0**	**100.0**
Mean-F	(3 × 3)	*98.8*	**100.0**	97.7	**100.0**
Median-F	(3 × 3)	94.5	95.7	*98.8*	**100.0**
Centered-C	20%	66.8	20.3	**98.4**	*94.5*
	50%	11.3	3.5	**66.0**	*48.0*
Edge-C	10%	18.8	55.5	*97.3*	**100.0**
	20%	6.3	28.5	*90.6*	**92.6**
Rotation	10°	66.4	36.7	**100.0**	*98.8*
	30°	8.2	5.9	*93.8*	**94.1**
	50°	5.1	2.3	*70.3*	**95.7**
Translation	(80, 50)	21.1	36.7	*98.4*	**98.8**
	(160, 100)	8.6	9.0	*88.3*	**94.1**
	(320, 200)	6.3	5.9	*70.3*	**91.8**
Scaling	0.5	*97.7*	92.2	87.5	**98.4**
	0.75	*96.9*	94.9	**100.0**	**100.0**
	1.5	*99.2*	98.8	**100.0**	**100.0**
	3	*98.8*	100.0	**100.0**	**100.0**
C-H-E		73.4	71.1	**96.5**	*75.0*
Gamma-C	0.8	89.8	*94.5*	**100.0**	**100.0**
Average		68.8	69.0	*91.2*	**95.7**

The bold data in the table represent the highest accuracy among the compared algorithms. The italicized data represent the second-highest accuracy.

Table 6: Robustness comparison with LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32] in holiday dataset.

Processing	Size	LDA_DCT (%)	DenseNet_DWT (%)	DCMH-CNN (%)	Proposed (%)
JPEG	Q (10)	93.0	96.9	*97.7*	**98.4**
	Q (50)	98.8	98.8	**100.0**	*99.6*
	Q (90)	**100.0**	**100.0**	**100.0**	*99.6*
Gauss-N	σ (0.001)	94.1	96.1	**99.2**	*98.8*
	σ (0.005)	91.4	95.7	*96.1*	**98.4**
S&P-N	σ (0.001)	98.0	*99.2*	**99.6**	**99.6**
	σ (0.005)	92.2	97.7	*98.8*	**99.2**
Speckle-N	σ (0.01)	93.0	96.1	98.0	**99.2**
	σ (0.05)	88.2	91.0	**91.4**	*89.5*
Gauss-F	(3 × 3)	**100.0**	**100.0**	**100.0**	*99.6*
Mean-F	(3 × 3)	**100.0**	**100.0**	**100.0**	*98.8*
Median-F	(3 × 3)	**100.0**	**100.0**	**100.0**	*99.6*
Centered-C	20%	31.1	19.5	**96.9**	*92.6*
	50%	9.8	5.9	*73.4*	**86.3**
Edge-C	10%	26.2	46.1	**95.3**	*92.6*
	20%	18.0	21.1	*80.9*	**89.8**
Rotation	10°	39.8	38.3	**98.8**	*94.5*
	30°	4.7	3.9	*72.7*	**91.4**
	50°	0.0	0.0	*49.2*	**90.2**

Table 6: Continued.

Processing	Size	LDA_DCT (%)	DenseNet_DWT (%)	DCMH-CNN (%)	Proposed (%)
Translation	(80, 50)	41.8	37.5	**98.8**	*98.0*
	(160, 100)	19.9	10.9	**98.8**	*97.3*
	(320, 200)	4.7	3.9	*93.8*	**96.1**
Scaling	0.5	**100.0**	**100.0**	**100.0**	*98.8*
	0.75	**100.0**	**100.0**	**100.0**	*99.2*
	1.5	**100.0**	**100.0**	**100.0**	*99.6*
	3	**100.0**	**100.0**	**100.0**	*99.6*
C-H-E		73.8	*85.9*	**95.7**	66.8
Gamma-C	0.8	94.1	98.0	**100.0**	*98.8*
Average		68.3	69.4	*94.1*	**95.4**

The bold data in the table represent the highest accuracy among the compared algorithms. The italicized data represent the second-highest accuracy.

Table 7: Robustness comparison with LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32] in Caltech-256 dataset.

Processing	Size	LDA_DCT (%)	DenseNet_DWT (%)	DCMH-CNN (%)	Proposed (%)
JPEG	Q (10)	91.0	*95.7*	71.5	**99.2**
	Q (50)	96.1	*98.0*	97.7	**99.6**
	Q (90)	97.7	98.8	**100.0**	*99.6*
Gauss-N	σ (0.001)	88.7	89.7	**100.0**	*99.6*
	σ (0.005)	84.4	87.9	*91.0*	**99.6**
S&P-N	σ (0.001)	90.6	87.9	**100.0**	*99.6*
	σ (0.005)	86.7	87.1	*98.4*	**99.6**
Speckle-N	σ (0.01)	90.6	88.3	*96.7*	**99.6**
	σ (0.05)	89.5	*89.8*	78.1	**98.4**
Gauss-F	(3×3)	98.8	**100.0**	**100.0**	*99.6*
Mean-F	(3×3)	*99.2*	**100.0**	94.9	*99.6*
Median-F	(3×3)	80.9	*93.0*	**99.6**	**99.6**
Centered-C	20%	49.2	6.3	**99.6**	*99.2*
	50%	5.9	3.5	*76.2*	**92.6**
Edge-C	10%	34.8	42.2	*98.1*	**99.6**
	20%	21.5	19.1	*93.4*	**99.6**
Rotation	10°	10.9	9.0	**99.6**	*99.2*
	30°	0.8	1.2	*86.7*	**96.1**
	50°	3.9	2.3	*64.8*	**94.5**
Translation	(80, 50)	18.4	19.9	*99.2*	**99.6**
	(160, 100)	4.3	4.3	**91.4**	**99.6**
	(320, 200)	3.5	2.0	*80.5*	**99.6**
Scaling	0.5	93.4	*99.2*	91.4	**99.6**
	0.75	93.8	*96.1*	**99.6**	**99.6**
	1.5	98.0	98.8	**100.0**	*99.6*
	3	*99.6*	**100.0**	**100.0**	*99.6*
C-H-E		60.5	68.0	**89.8**	*79.7*
Gamma-C	0.8	90.2	92.6	**100.0**	*99.6*
Average		63.7	63.6	*92.8*	**98.2**

The bold data in the table represent the highest accuracy among the compared algorithms. The italicized data represent the second-highest accuracy.

algorithm to the DCMH-CNN [32] algorithm whose average accuracy is the second highest. In rotation attacks, the DCMH-CNN [32] algorithm has a fluctuation range of 29.7%–49.6%, while our algorithm exhibits a much smaller fluctuation range of only 4.2%–4.7%. In translation attacks, the fluctuation range of the DCMH-CNN [32] algorithm is 5.0%–28.1%, whereas our algorithm has a fluctuation range of 0%–7%. Notably, the proposed algorithm exhibits significantly higher robustness than the compared algorithms when facing rotation attacks.

From the tables, we can observe that the proposed algorithm performs well against noise, filtering compression, rotation, and scaling attacks. This is because SURFs possess several characteristics:

(1) Scale invariance: SURF employs multiscale analysis, allowing it to detect feature points at different scales. This provides robustness against scale variations caused by noise, filtering, and compression.

(2) Rotation invariance: SURF ensures matching of corresponding feature points even in the presence of object rotation through orientation assignment steps. This contributes to the algorithm's robustness when facing rotation attacks.

(3) Robust feature descriptors: SURF calculates feature descriptors based on local gradients and orientation information, rather than relying on absolute brightness values. This reduces the impact of noise, filtering, and compression on feature extraction, enhancing the robustness of SURF against these attacks.

Overall, the scale invariance, rotation invariance, robust feature descriptors, and adaptability to geometric transformations make SURF retrieval perform well against noise, filtering and compression, rotation, and scaling attacks.

The accuracy of the ImageNet database in the 50% Centered-C (center cropping) attack is lower than that of the Caltech-256 and Holiday databases. This is because, after the 50% Centered-C, only a single background color is left for most of the images in the ImageNet database, resulting in similar SURFs for the cropped images. This similarity leads to incorrect matching of original images in the CID, resulting in lower accuracy. However, for the Caltech-256 and Holiday image databases, there are still sufficient edge features available for matching original CID images even after 50% center cropping, resulting in higher accuracy. In summary, SURFs primarily focus on texture and structure of images and are not sensitive to single-color variations.

While the proposed algorithm exhibits low accuracy against C-H-E (color histogram equalization) attack, primarily due to the redistribution of the histogram which alters the color distribution and contrast of the images, this results in the loss or confusion of local structural information during the SURF extraction process, further reducing accuracy.

To examine the impact of different image sizes on the accuracy of secret information extraction, we conducted experiments on image databases using various sizes in addition to performing the same resizing as LDA_DCT [18], DenseNet_DWT [19], and DCMH-CNN [32] methods. Specifically, we conducted experiments by resizing the ImageNet image database to 256×256 and 512×512, the Caltech256 image database to 256×256 and 512×512, and the Holidays image database to 128×128 and 256×256. We compared the average accuracy of secret information extraction from the images after various attacks. The experimental results are shown in Figure 10.

From the plotted curves in Figure 10, we can observe that the average accuracy of secret information extraction fluctuates within a range of 2.5% as the image size changes. This indicates that our algorithm demonstrates relatively stable average accuracy when the image size varies, even with the same content in the images.

4.4. Security

4.4.1. Steganalysis Security Analysis. In the proposed algorithm, the carrier images from the CIDs remain unmodified; it allows evasion of detection by all steganalysis algorithms, which is the superior characteristic of the coverless image steganography compared with the traditional image steganography.

4.4.2. Transmission Security Analysis. In coverless steganographic algorithms, it is true that traditional steganalysis techniques may not detect the secret information carried within the transmitted carrier images. However, this does not imply that coverless steganography algorithms are completely secure. Existing coverless steganography algorithms, such as LDA_DCT [18] and DenseNet_DWT [19], often require the transmission of additional information regarding the positions of the image blocks. The amount of additional information needs to be sent along with each cover image. The longer the additional information is, the easier it is for the listener to perceive. Once detected, it poses significant security risks. For example, the listener can modify the additional information, leading to incorrect extraction of the secret message, or delete the additional information, making the secret message unrecoverable.

DCMH-CNN [32] does not require the transmission of additional position information, but it requires both communicating parties to use the same network for feature extraction. This necessitates the transmission of a significant amount of network parameters and network structures, introducing certain security risks. Furthermore, when the length of the secret information is much smaller than the transmitted additional information (such as network parameters), it not only loses the purpose of hiding information but also raises suspicions from observers.

Besides transmission security, additional information also brings additional transmission costs. To better illustrate the impact of additional information on transmission cost, we define a ratio of additional information to capacity, which is calculated as follows:

$$\text{Ratio} = \frac{L_a}{L}, \tag{12}$$

where L_a represents the length of the additional information which is converted to binary. A lower ratio indicates lower transmission costs. For example, if the additional information is r, the corresponding length would be $log_2 r$ bits. After investigation, we found that the length of the additional information is unrelated to the length of the secret information but is related to the hash length L. The specific comparison results are shown in Table 8.

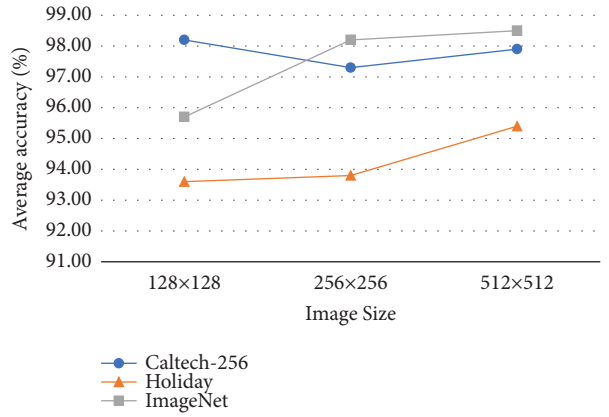

Figure 10: Different image sizes correspond to accuracy.

Table 8: Comparison of additional transmission cost.

Method	Additional considerations	Length of secret/bits	Length of additional information/bits	Ratio
LDA_DCT [18]	Position information (*rx*, *ry*)	1~15	0~8	0~8
DenseNet_DWT [19]	Position information (*rx*, *ry*)	1~15	0~8	0~8
DCMH-CNN [32]	Number of clustering *K*	1~16	4	0.25~4
Our	None	1~16	0	0

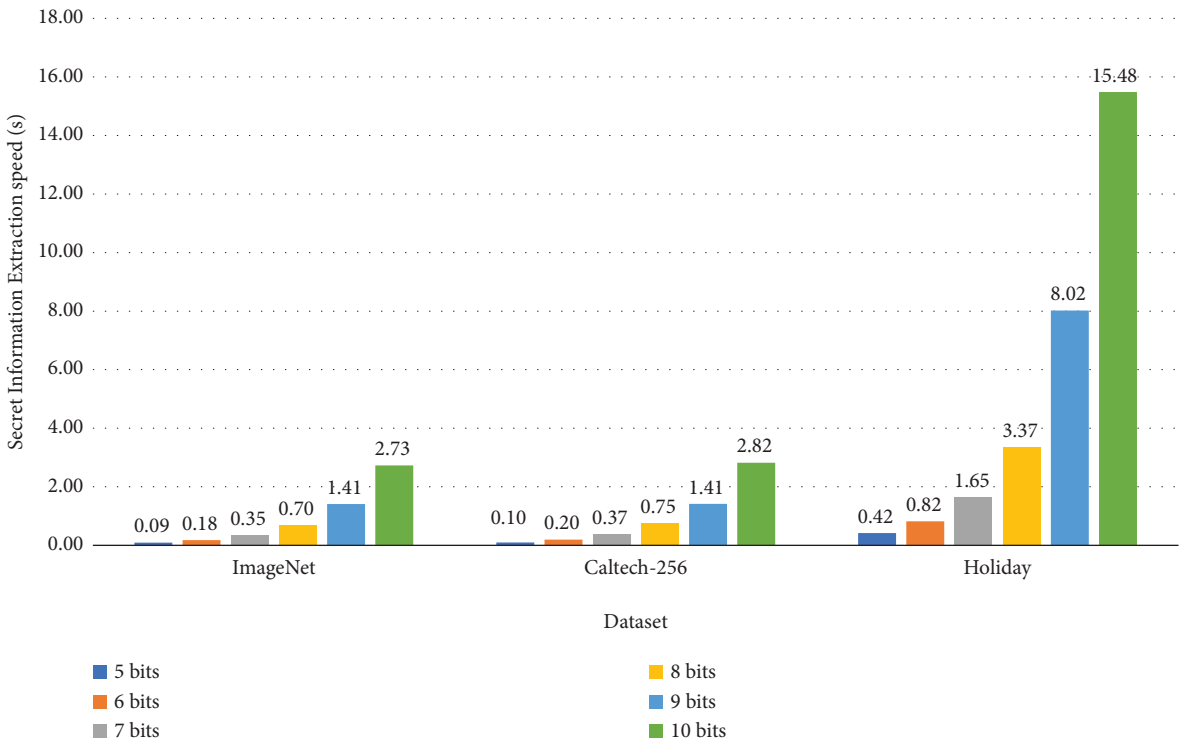

Figure 11: Password information extraction efficiency.

To better compare the additional transmission cost, we have considered the hash sequence with various lengths. In the comparison, we can see that depending on the position information needed to transmit the secret information, the required additional information length varies for the methods LDA_DCT [18] and DenseNet_DWT [19]. If the image block size varies during block segmentation, the required additional information length differs. The smaller the

Table 9: Summary of performance.

Method	Dataset (Image size)	Preprocessing	Methods	Result
LDA_DCT [18]	INRIA holiday (512 × 512) ImageNet (128 × 128) Caltech-256 (128 × 128)	Image block scanning DCT value	Generative-based; generate corresponding hash sequences based on image DCT values; both communication parties use the same feature	Low capacity and low robustness. It has the security risk of transmitting the same image repeatedly and amount of additional information
DenseNet_DWT [19]	INRIA holiday (512 × 512) ImageNet (128 × 128) Caltech-256 (128 × 128)	Image block scanning DWT value	Generative-based; generate corresponding hash sequences based on image DWT values; both communication parties use the same feature	Low capacity and low robustness. It has the security risk of transmitting the same image repeatedly and amount of additional information
DCMH-CNN [32]	INRIA holiday (512 × 512) ImageNet (128 × 128) Caltech-256 (128 × 128)	Extract high-dimensional features for sorting	Mapping-based; extract high-dimensional features using a network and use K-means clustering to form a CID; sort the high-dimensional features and construct mapping rules; both communication parties use the same feature	High capacity and strong robustness. But it has the security risk of transmitting the same image repeatedly and additional information
Our	INRIA holiday (128 × 128, 256 × 256, 512 × 512) ImageNet (128 × 128, 256 × 256, 512 × 512) Caltech-256 (128 × 128, 256 × 256, 512 × 512)	Calculate average pixel value for sorting	Mapping-based; generate a random sequence to form a CID; construct mapping rules; only the receiver uses SURF features	High capacity and strongest robustness. There is no security risk of repeatedly transmitting the same image and additional information

selected segmented image blocks, the more position information is needed, resulting in a greater amount of additional information to be sent. For example, if the segmented block size is 16×16, then 8 bits of additional information length is needed to provide to the receiver. If the hash sequence length is set as 1 bit, then the ratio is 8. Compared to the extensive position information required to be sent in generative methods, the DCMH-CNN [32] method only needs to transmit the number of clustering K to the receiver, with a fixed additional information length of 4 bits. The transmission cost is extremely high for all three methods, especially when choosing a smaller hash sequence length. In addition, the table does not list the requirements of network consistency between the sender and receiver for the DCMH-CNN [32], but it also has significant transmission security and cost concerns.

Another important security issue is that when the secret information is divided into n segments, there might be some repeated segments. In existing steganography methods, it requires sending the same image multiple times when facing repeated segments, which will cause suspicion to listeners. However, in our method, we have established multiple sub-CIDs, which allow us to avoid sending the same image repeatedly. Instead, when a secret information segment is repeated, we can select the corresponding image from another CID subdatabase in a sequential manner. This operation improves transmission security by minimizing the risk of detection or interception. It also provides better efficiency in utilizing the available images in the CID subdatabases, as we can choose different images for each repeated segment, increasing the diversity of the transmitted data and enhancing the security of the steganographic communication.

4.5. Secret Information Extraction Efficiency. The speed of extracting secret information from steganographic images is crucial for the efficiency and practical application of steganography algorithms. Fast and accurate extraction of secret information can greatly reduce time costs and provide strong support for timely decision-making and mitigating potential risks. The generative-based image steganography algorithm relies primarily on generating a hash sequence from image features, and the speed of extracting secret information depends on the speed of generating the hash sequence. On the other hand, the mapping-based image steganography algorithm, upon receiving an image, searches the original image in the CID and retrieves the corresponding secret information based on the mapping rules. The speed of extracting secret information is determined by finding the corresponding original image in the CID. In Figure 11, we have calculated the extraction speed of secret information under different capacities.

The x-axis in Figure 11 represents three image databases, while the y-axis represents the time of extracting the secret information from a stego image, which is measured in seconds. Each bar in Figure 11, from left to right, represents the time required with capacities ranging from 5 to 10 bits. From the graph, it can be observed that as the capacity increases, the extraction time of secret information also increases. This is because, with the increase in capacity, the number of required sub-CIDs also increases. For every 1-bit increase in capacity, the number of images in the sub-CIDs doubles, resulting in an approximately doubled retrieval time. In terms of overall extraction speed, ImageNet and Caltech-256 image databases perform better. Even when the capacity of each stego image reaches 10 bits, the extraction speed remains below 3 seconds. However, in the case of the Holiday image database, the time required to extract information is much higher compared to the other two databases. This is because the ImageNet and Caltech-256 image databases have image sizes of 128×128, while the Holiday image database has an image size of 512×512. The larger image size leads to a higher number of pixels that need to be computed for the extraction of SURFs. Overall, the speed of secret information extraction in the proposed method is relatively fast.

4.6. Summary of Performance. In this section, we summarized all the methods in Table 9 in terms of dataset (image size), preprocessing, methods, and results.

From Table 9, we can see that our experiment used the same dataset as other methods, but we investigated the impact of images of various sizes on the experiments for the proposed algorithm. In terms of preprocessing, LDA_DCT [14] and DenseNet_DWT [15] are generative-based methods that scan the DCT values and DWT values, respectively, while the DCMH-CNN [28] and the proposed method belong to mapping-based methods, where images are sorted according to predefined rules. In terms of methods, LDA_DCT [14] and DenseNet_DWT [15] generate corresponding hash sequences based on DCT and DWT values, respectively. The DCMH-CNN [28] and the proposed method use constructed mapping relationships to generate hash sequences for images. In terms of results, our capacity is the same as the latest DCMH-CNN [28] method, ranking the highest among the four methods. In terms of robustness, the proposed method outperforms the other four methods. In terms of security, unlike the other four methods, the security risk of transmitting the same image repeatedly and additional information is avoided in the proposed method.

5. Conclusion

In this paper, a robust coverless image steganography algorithm based on the SURF image retrieval feature is proposed. Multiple CIDs are constructed from a public dataset using random seeds; then, a mapping rule is designed to establish a one-to-one correspondence between CID images and hash sequences. The receiver utilizes SURF to retrieve the original image of the received one, and then get the secret information it represents. The proposed method addresses several important security issues in both generative and mapping-based coverless image steganography algorithms, which avoids transmitting a large amount of additional information and repetitive transmission of the same image. Experimental results demonstrate that the proposed method is capable of resisting most malicious

image attacks and exhibits stronger robustness compared to other advanced techniques. In future, we will further improve the robustness of the proposed method, especially the robustness against CHE and Center-C cropping attacks which are difficult for all the coverless image steganography.

Data Availability

No underlying data were collected or produced in this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62205092).

References

[1] R. Zhang, S. Dong, and J. Liu, "Invisible steganography via generative adversarial networks," *Multimedia Tools and Applications*, vol. 78, pp. 8559–8575, 2019.

[2] X. Duan, K. Jia, B. Li, D. Guo, E. Zhang, and C. Qin, "Reversible image steganography scheme based on a U-net structure," *IEEE Access*, vol. 7, pp. 9314–9323, 2019.

[3] O. Elharrouss, N. Almaadeed, and S. Al-Maadeed, "An image steganography approach based on k-least significant bits (k-LSB)," in *Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, Doha, Qatar, February 2020.

[4] X. Duan, W. Wang, N. Liu, D. Yue, Z. Xie, and C. Qin, "StegoPNet: image steganography with generalization ability based on pyramid pooling module," *IEEE Access*, vol. 8, pp. 195253–195262, 2020.

[5] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.

[6] X. Zhao, C. Yang, and F. Liu, "On the sharing-based model of steganography," in *Proceedings of the Digital Forensics and Watermarking: 19th International Workshop, IWDW 2020*, pp. 94–105, Melbourne, Australia, November 2020.

[7] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 2995–3004, 2019.

[8] S. Jing-yu, C. Hong, W. Gang, G. Zi-bo, and H. Zhang, "FPGA image encryption-steganography using a novel chaotic system with line equilibria," *Digital Signal Processing*, vol. 134, 2023.

[9] B. Wei, X. Duan, and H. Nam, "Image steganography with deep learning networks," in *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, October 2022.

[10] L. Liu, L. Tang, and W. Zheng, "Lossless image steganography based on invertible neural networks," *Entropy*, vol. 24, no. 12, p. 1762, 2022.

[11] N. Subramanian, I. Cheheb, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "End-to-End image steganography using deep convolutional autoencoders," *IEEE Access*, vol. 9, Article ID 135585, 2021.

[12] Y. Lan, F. Shang, J. Yang, X. Kang, and E. Li, "Robust image steganography: hiding messages in frequency coefficients," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023.

[13] Z. Yang, K. Chen, K. Zeng, W. Zhang, and N. Yu, "Provably secure robust image steganography," *IEEE Transactions on Multimedia*, vol. 26, pp. 5040–5053, 2024.

[14] K. Karampidis, E. Kavallieratou, and G. Papadourakis, "A review of image steganalysis techniques for digital forensics," *Journal of Information Security and Applications*, vol. 40, pp. 217–235, 2018.

[15] D. Megías and D. Lerch-Hostalot, "Subsequent embedding in targeted image steganalysis: theoretical framework and practical applications," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 1403–1421, 2023.

[16] Z. Lin, Y. Huang, and J. Wang, "RNN-SM: fast steganalysis of VoIP streams using recurrent neural network," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1854–1868, 2018.

[17] Z. Zhou, H. Sun, R. Harit, X. Chen, and X. Sun, "Coverless image steganography without embedding," *Cloud Computing and Security*, vol. 1, pp. 123–132, 2015.

[18] X. Zhang, F. Peng, and M. Long, "Robust coverless image steganography based on DCT and LDA topic classification," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3223–3238, 2018.

[19] Q. Liu, X. Xiang, J. Qin, Y. Tan, J. Tan, and Y. Luo, "Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping," *Knowledge-Based Systems*, vol. 192, 2020.

[20] S. Zheng, L. Wang, B. Ling, and D. Hu, "Coverless information hiding based on robust image hashing," *Intelligent Computing Methodologies*, vol. 13, pp. 536–547, 2017.

[21] C. Yuan, Z. Xia, and X. Sun, "Coverless image steganography based on SIFT and BOF," *Journal of Internet Technology*, vol. 18, pp. 435–442, 2017.

[22] Y. Luo, J. Qin, X. Xiang, and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2779–2791, 2021.

[23] Q. Liu, X. Xiang, J. Qin, Y. Tan, and Q. Zhang, "A robust coverless steganography scheme using camouflage image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 4038–4051, 2022.

[24] N. A. Karim, S. A. Ali, and M. J. Jawad, "A coverless image steganography based on robust image wavelet hashing," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 6, pp. 1317–1325, 2022.

[25] X. Chen, A. Qiu, X. Sun, S. Wang, and G. Wei, "A high-capacity coverless image steganography method based on double-level index and block matching," *Mathematical Biosciences and Engineering: MBE*, vol. 16, no. 5, pp. 4708–4722, 2019.

[26] D. Hu, L. Wang, W. Jiang, S. Zheng, and B. Li, "A novel image steganography method via deep convolutional generative adversarial networks," *IEEE Access*, vol. 6, Article ID 38303, 2018.

[27] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu, and L. Xiang, "Coverless real-time image information hiding based on image block matching and dense convolutional network," *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125–135, 2020.

[28] S. Biswas, S. Debnath, and R. K. Mohapatra, "Coverless image steganography based on DWT approximation and pixel

intensity averaging," in *Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, April 2023.

[29] Z. Zhou, Y. Mu, and Q. M. J. Wu, "Coverless image steganography using partial-duplicate image retrieval," *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2018.

[30] Q. Li, X. Wang, X. Wang, B. Ma, C. Wang, and Y. Shi, "An encrypted coverless information hiding method based on generative models," *Information Sciences*, vol. 553, pp. 19–30, 2021.

[31] X. Chen, Z. Zhang, A. Qiu, Z. Xia, and N. Xiong, "Novel coverless steganography method based on image selection and StarGAN," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 219–230, 2022.

[32] L. Zou, J. Li, W. Wan, Q. M. J. Wu, and J. Sun, "Robust coverless image steganography based on neglected coverless image dataset construction," *IEEE Transactions on Multimedia*, vol. 25, pp. 5552–5564, 2023.

[33] Q. Jiang and W. Li, "Deep cross-modal hashing," in *Proceedings of the IEEE conference on computer vision and pattern Recognition (CVPR)*, pp. 3270–3278, Silver Spring, MD, USA, February 2017.

[34] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[35] Y. Luo, J. Qin, X. Xiang, and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2779–2791, 2021.