

High-Capacity Reversible Data Hiding in Encrypted Hyperspectral Images Using MSB Prediction and Arithmetic Coding

Zhe-Min Yeh¹, Chia-Chen Lin^{2,*}, Yijie Lin¹, Ching-Chun Chang³, Chin-Chen Chang^{1,*}

¹Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan
{m1271774, p1263670, ccc}@o365.fcu.edu.tw

²Department of Computer Science and Information Engineering,
National Chin-Yi University of Technology, Taichung 411, Taiwan
Ally.cclin@ncut.edu.tw

³Information and Communication Security Research Center, Feng Chia University,
NO.100 Wenhwa Rd., Seatwen, Taichung 407, Taiwan, Republic of China
ccc@fcu.edu.tw

*Corresponding authors: Chia-Chen Lin, Chin-Chen Chang

Received April 30, 2025, revised May 24, 2025, accepted May 25, 2025.

ABSTRACT. *With the increasing computational power and advancements in hardware technologies, processing large-scale digital datasets—such as hyperspectral images—has become more accessible for both individuals and organizations. However, the high cost associated with acquiring hyperspectral data continues to pose challenges, especially regarding the protection of intellectual property. Reversible data hiding (RDH) presents a viable strategy to embed watermarks into multimedia content to ensure copyright preservation. Nevertheless, traditional RDH approaches, originally intended for natural images, often fail to exploit the rich inter-band correlations characteristic of hyperspectral imagery, limiting their efficiency in this domain. To address this gap, we introduce a specialized Reversible Data Hiding in Encrypted Hyperspectral Images (RDHEI) technique. This method capitalizes on the intrinsic spectral similarity between adjacent bands to enhance prediction accuracy. By integrating Most Significant Bit (MSB) analysis with arithmetic encoding, the proposed scheme supports high-capacity embedding while maintaining image fidelity. The framework begins by dividing MSBs according to diverse embedding requirements. It then utilizes arithmetic encoding based on the observed MSB variance to open up additional space for data insertion. Image encryption is applied before final-stage embedding. Performance evaluations reveal that our method significantly outperforms existing solutions, achieving an average embedding rate of 3.76 bits per pixel per band (bpppb), in contrast to the 0.36 bpppb reported in Zhang et al.'s approach.*

Keywords: reversible data hiding in the encrypted image; reversible data hiding in the encrypted hyperspectral image; distortion-free; hyperspectral images; steganography; high embedding rate

1. **Introduction.** Data hiding has gained significant attention for its ability to embed confidential information into digital media, such as images, audio, and videos while ensuring imperceptibility. This capability is particularly critical in sensitive domains like military operations, geosciences, and medical research, where the lossless recovery of the original media is essential. Reversible Data Hiding (RDH) plays a vital role in this field by enabling the embedding of secret information while ensuring the exact restoration of the original content. RDH techniques encompass a range of approaches, including compression-based methods [1, 2, 3, 4, 5], spatial domain techniques [6, 7, 8, 9, 10, 11, 12, 13, 14, 15], and encrypted domain strategies [16, 17, 18, 27, 28, 29, 30]. With the continuous emergence of diverse image formats, novel RDH methods have been developed for specialized media, such as high dynamic range (HDR) images [19]. Among these specialized media, hyperspectral images (HSI) are particularly noteworthy due to their unique characteristics and the specific challenges they pose.

Hyperspectral imaging (HSI) captures hundreds of narrow spectral bands, providing detailed spectral information for each pixel. This capability makes HSI invaluable in applications such as environmental monitoring, agriculture, and defense [20]. However, its unique features, such as high dimensionality, large size, and strong correlation between adjacent spectral bands pose significant challenges for data hiding. Traditional data hiding techniques [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 19, 21, 22, 27, 28, 29, 30] developed for natural images are often unsuitable for HSI. Moreover, the reversible data hiding in encrypted images (RDHEI) technique, which enables secret data embedding within encrypted images while allowing for the lossless recovery of both the embedded data and the original image [23], often fails to leverage the unique properties of HSI, such as its inherent redundancy. In addition to traditional encryption methods, hyperspectral image encryption techniques that rely on complex transformations, such as the Fourier transform, have been explored to enhance security [24]. These approaches use the frequency domain to obscure spatial information, making it harder to hide data in encrypted images. However, these techniques often introduce significant computational complexity and may not fully take advantage of the spectral redundancy in HSI for efficient reversible data hiding. Building on these challenges, our proposed scheme seeks to overcome these limitations by utilizing the redundancy in the most significant bits (MSBs) and the strong spectral correlation in HSI, enabling efficient and secure data embedding while ensuring lossless recovery.

The applications of hyperspectral imaging (HSI) extend beyond the previously mentioned fields, encompassing geology and mining, food quality control in the food and beverage industry, and art conservation. Both copyright protection and confidentiality are critical concerns in these areas. Therefore, exploring the potential to expand reversible data hiding in encrypted images (RDHEI) to HSI has become increasingly important. Although the format of HSI differs significantly from traditional images, its high spectral redundancy between adjacent bands provides predictive patterns that can be effectively utilized for data embedding. Additionally, the redundancy in the most significant bits (MSBs) of pixels—particularly in higher spectral bands where these bits often carry minimal or no valuable information—presents opportunities for low-impact data embedding [25].

To enhance solutions for confidentiality and copyright protection in HSIs, this research introduces a novel RDHEI scheme specifically designed for HSI. The proposed approach leverages MSB redundancy and the strong correlation between adjacent spectral bands to enable efficient and secure data embedding. By incorporating a dynamic encoding mechanism, the scheme adjusts the embedding process based on variations in MSBs of pixels, ensuring both lossless recovery of the embedded data and minimal image distortion. This

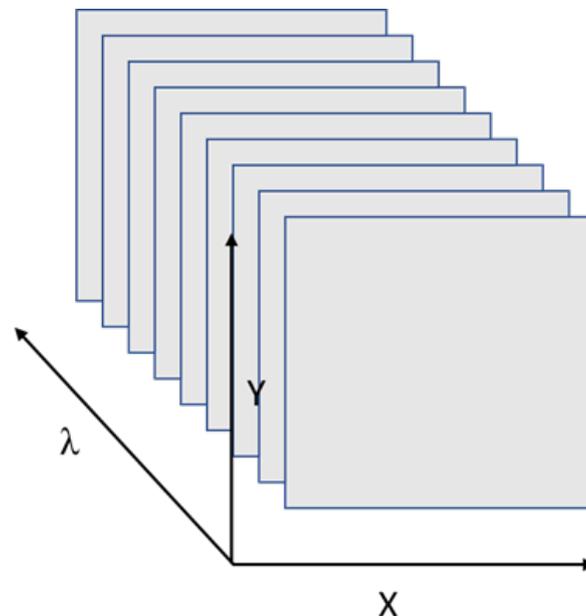
method is particularly advantageous for applications demanding high embedding capacity, robust security, and data integrity. The effectiveness and efficiency of the proposed scheme are validated through experiments conducted on standard hyperspectral image datasets, demonstrating its practical contributions and potential.

The remainder of the paper is structured as follows: Section 2 provides an overview of hyperspectral imaging (HSI) and a review of existing reversible data hiding (RDH) techniques tailored for HSI [26]. Section 3 details the proposed scheme, including its encryption process, embedding strategy, and data retrieval mechanisms. Section 4 presents experimental results using standard hyperspectral datasets, demonstrating the scheme's effectiveness and efficiency. Finally, Section 5 concludes the paper by summarizing the key contributions and exploring potential directions for future research.

2. Related Work.

2.1. Hyperspectral Imaging. In hyperspectral imaging (HSI), each pixel in an image spectrum represents a continuous range of wavelengths captured across the electromagnetic spectrum. Unlike traditional RGB images, which store information in three channels—red, green, and blue—HSI provides significantly higher spectral resolution by dividing the light spectrum into numerous contiguous bands. Depending on the sensor's resolution, these bands can range from tens to hundreds, allowing for precise material identification based on unique spectral signatures.

The spectral information for each pixel in a hyperspectral image (HSI) is represented as a data cube, commonly referred to as a hypercube, as illustrated in Figure 1. This hypercube consists of three dimensions: two spatial dimensions (x and y) and one spectral dimension (λ). The spectral dimension (λ) represents the reflectance or radiance intensity at each specific wavelength. Given the large volume of data, this hypercube demands efficient storage and processing techniques, especially in bandwidth-limited environments.



2

FIGURE 1. HSI data cube structure, called hypercube

The radiance or reflectance at each spectral band is usually stored as a 16-bit value to ensure high precision. As a result, the pixel values in an HSI generally range from 0 to 65535 ($= 2^{16} - 1$). This high precision allows the detection of subtle differences in

spectral characteristics, which is critical for applications like vegetation analysis, mineral identification, and target detection in remote sensing.

2.2. Zhang et al.'s Scheme. In 2024, Zhang et al. introduced an RDH scheme tailored for hyperspectral images compressed using absolute moment block truncation coding (AMBTC) [26]. Their scheme divides the bitmap into 2×2 or 4×4 non-overlapping blocks and then leverages bitmap prediction, an adaptive embedding order, and a dynamical embedding scheme to achieve reversible embedding with minimized distortion and high embedding capacity.

The embedding algorithm is shown as follows:

Input: Hyperspectral image compressed using AMBTC: $C_{i,k} = \{h_{i,k}, l_{i,k}, S_{i,k}\}$, secret information to embed. Here, $h_{i,k}$ represents the high quantization value of the i th block in the k th band, $l_{i,k}$ represents the low quantization value, and $S_{i,k}$ represents the bitmap of the i th block in the k th band.

Output: Stego AMBTC codes with embedded secret information and auxiliary information.

Step 1: Predict the bitmap $S_{i,k}$ for the current band using the previous band's bitmap $S_{i,k-1}$. Measure the prediction accuracy using Hamming Distance:

$$HD_{i,k} = \sum_{j=1}^{n_1 \times n_2} |s_{i,j,k} - \hat{s}_{i,j,k}| \quad (1)$$

Here, n_1 and n_2 represent the height and width of the block, respectively. $s_{i,j,k}$ represents the j th pixel value in $S_{i,k}$, while $\hat{s}_{i,j,k}$ represents the predicted value of $s_{i,j,k}$ based on $S_{i,k-1}$.

Step 2: Rank blocks by complexity to prioritize low-complexity blocks for embedding:

$$\text{Complexity}_{i,k} = (h_{i,k} - l_{i,k}) \cdot (HD_{i,k} + \text{bias}) \cdot (\text{Disorderliness} + \text{bias}) \quad (2)$$

Here, bias is a small positive constant used to prevent zero values, while Disorderliness is an indicator of pixel variations within the block, used to assess its suitability for embedding.

Step 3: Select one embedding strategy from three embedding strategies based on the predetermined block size, i.e., 2×2 or 4×4 , the required embedding capacity, and the concerned distortion constraints:

- Type 1: Divide the bitmap into the smaller non-overlapping block size, such as 2×2 , and embed secrets when the HD is 0 or 1 to reduce the distortion. If HD is 0, replace value of $S_{i,k}$ with the embedding secret data. Otherwise, record the location information as the auxiliary information, embed the secret data and switch the two quantization values.
- Type 2: Divide the bitmap into the smaller non-overlapping block size, such as 2×2 , and embed secrets when the Hamming distance is 0 to reduce the distortion. It is noted Type 2 is suitable for the hiding capacity is the least and the minimal distortion is preferred. Since Type 2 only embeds secret data for HD is 0; therefore, no auxiliary information is required.
- Type 3: Divide the bitmap into the non-overlapping block size, such as 4×4 . It is noted Type 3 is suitable for the larger hiding capacity and the possible distortion is acceptable. To achieve this objective, Difference Expansion (DE) is adopted to embed auxiliary information in quantization values and secret bits in the bitmap.

Step 4: According to the selected embedding strategy to modify the bitmap and/or quantization values to embed secret bits and auxiliary information. Record auxiliary

information, such as embedding scheme, block size, and position of last embedded bit, to ensure reversibility.

Step 5: Save the modified AMBTC codes $\{h'_{i,k}, l'_{i,k}, S'_{i,k}\}$ with embedded secret information.

The extracting algorithm is shown as follows:

Input: Stego AMBTC codes $\{h'_{i,k}, l'_{i,k}, S'_{i,k}\}$

Output: Recovered original AMBTC codes $\{h_{i,k}, l_{i,k}, S_{i,k}\}$, extracted secret information.

Step 1: Extract auxiliary information embedded in quantization values, including the embedding scheme used, block size, and position of the last embedded bit according to the embedding strategy selected in the embedding algorithm.

- Type 1: Check if two quantization values have been switched to decide whether secret data are carried. If HD is 0, extract the hidden data as the secret data. Otherwise, read the auxiliary information first, and the rest of the data are the hidden secret data.
- Type 2: Check if two quantization values have been switched to decide whether secret data are carried. If two quantization values have been switched, read the bitmap values as the secret data.
- Type 3: Extract the hidden data according to the DE extraction operation.

Step 2: Reverse bitmap modifications and restore $S_{i,k}$ and restore quantization values $h_{i,k}$ and $l_{i,k}$.

Step 3: Recover secret bits from the modified bitmaps and auxiliary data and reorder the extracted bits based on the original complexity ranking.

Step 4: Compare the recovered AMBTC codes $\{h_{i,k}, l_{i,k}, S_{i,k}\}$ with the original codes to confirm no information loss.

2.3. Arithmetic Coding. Among compression-based Reversible Data Hiding (RDH) methods, Arithmetic Coding (AC) is a widely used lossless compression technique that effectively reduces data redundancy and enhances embedding efficiency. Arithmetic coding encodes an entire input message by mapping it to a real number within the range $[0, 1)$, ensuring that the encoded length is proportional to the probability of symbol occurrence. This technique has found extensive use in image compression (e.g., JPEG2000) and information hiding. For instance, [26] proposed an RDH method based on arithmetic coding that leverages the residual redundancy in the encoding process to embed additional information, thereby achieving a higher embedding capacity.

Moreover, in the domain of encrypted images, arithmetic coding has been employed to strengthen the security of RDH methods. Research has demonstrated that arithmetic coding can be combined with encryption techniques to ensure the lossless recovery of the image while improving the imperceptibility and robustness of data hiding. For example, [27] introduced a Reversible Data Hiding in Encrypted Images (RDHEI) method based on arithmetic coding, which performs arithmetic coding within the encrypted domain and uses frequency prediction to improve data embedding efficiency. Despite these advances, these methods still face challenges in terms of computational complexity, particularly in high-dimensional images such as hyperspectral images (HSI), where balancing computational cost and embedding efficiency remains an ongoing research challenge.

To further illustrate how arithmetic coding works, consider the following example: Arithmetic coding encodes an input message as a fractional number within the range $[0, 1)$. The symbol probabilities for the symbols “A”, “B”, and “C” used in this demonstration are listed in Table 1:

TABLE 1. Probability Intervals for Arithmetic Coding

Symbol	Probability Range
A	[0.0, 0.5)
B	[0.5, 0.8)
C	[0.8, 1.0)

Suppose the sequence “ABAC” is encoded using Arithmetic coding. The encoding process proceeds as follows:

Step 1. **Initialization:** Set the initial range as $[0.0, 1.0)$. And then, the initial range is set as $[low = 0.0, high = 1.0]$

Step 2. **Processing “A”:** Symbol “A” has a probability of 0.5, so the range for “A” is $[0.0, 0.5)$. New range after processing “A” is set as $[0.0, 0.5)$

Step 3. **Processing “B”:** Symbol “B” has a probability of 0.3, so the range for “B” is $[0.5 \times 0.0, 0.5 \times 0.3]$ and $[0.5 \times 0.3, 0.5 \times 0.6]$. Finally, new range after processing “B” is set as $[0.25, 0.4)$

Step 4. **Processing “A”:** Symbol “A” has a probability of 0.5, so the range for “A” is $[0.0, 0.5)$. New range after processing “A” within the interval $[0.25, 0.4]$ is computed as $[0.25 + 0.15 \times 0.0, 0.25 + 0.15 \times 0.5)$. The final new range is set as $[0.25, 0.325)$

Step 5. **Processing “C”:** Symbol “C” has a probability of 0.2, so the range for “C” is $[0.8, 1.0)$. New range after processing “C” within the interval $[0.25, 0.325]$ is computed as $[0.25 + 0.075 \times 0.8, 0.25 + 0.075 \times 1.0)$. Finally, the new range is set as $[0.31, 0.325)$

Step 6. **Final Encoding:** The final encoded value lies within the range $[0.31, 0.325)$. Any number in this interval (e.g., 0.315) can serve as the compressed representation of the sequence “ABAC”.

Thus, the sequence “ABAC” has been encoded efficiently into the number 0.315 (or any value in the interval $[0.31, 0.325)$), representing the entire message.

3. Proposed scheme. This section provides a detailed explanation of our approach, which integrates an advanced embedding mechanism based on MSB prediction, an innovative data hiding technique for hyperspectral images, and a dynamic embedding strategy designed to meet varying requirements. It thoroughly describes the proposed scheme, including the embedding and extraction processes, along with an overview of the scheme’s framework. Illustrated in Figure 2, the scheme consists of three stages: content-owner, data-hider, and receiver.

In the content-owner stage, the MSBs of pixels in the HSI are extracted and encoded, while the image itself is encrypted using an image encryption key. During the data-hider stage, secret data is encrypted with a data encryption key and embedded into the encrypted image, resulting in an embedded image containing the secret data. Finally, in the receiver stage, the HSI is recovered using the image decryption key, and the embedded secret data is retrieved using the data decryption key.

3.1. Hyperspectral imaging. The core concept of our data hiding strategy involves creating reversible space by compressing the MSBs of pixels in the HSI through encoding, enabling the embedding of secret data bits S into the HSI. The secret data S is represented as $S = (S_0, S_1, \dots, S_r)$, and each S_i is a randomly generated secret data bit such that $S_i \in \{0, 1\}$ and $0 \leq i \leq r$.

The HSI I is defined with dimensions as $W \times H \times B$, where W is the width, H is the height, and B represents the number of spectral bands in the HSI. Figure 3 provides examples of HSI used in our scheme, simulated using a data cube with dimensions $2 \times 3 \times 4$.

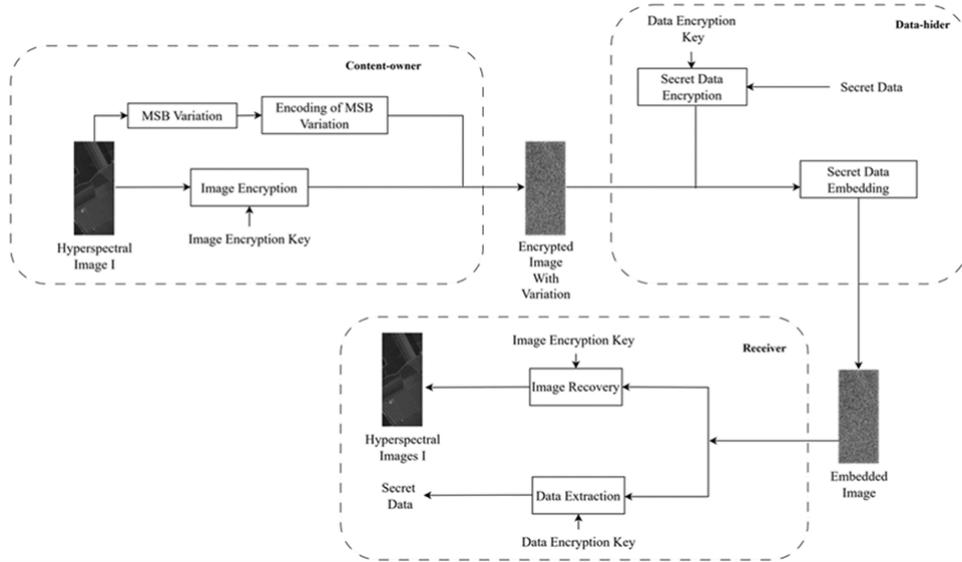


FIGURE 2. Overview of the proposed scheme

Each pixel value across the bands is displayed in both decimal and binary formats, with the fourth MSB in the binary representation highlighted in red for better visualization.

Band1		Band2	
P_1 (4075) / 0000111111101 011	P_2 (4438) / 0001000101010 110	P_1 (4264) / 0001000010101 000	P_2 (5001) / 0001001110001 001
P_3 (3779) / 0000111011000 011	P_4 (4435) / 0001000101010 011	P_3 (5357) / 0001010011101 101	P_4 (4988) / 0001001101111 100
P_5 (4077) / 0000111111101 101	P_6 (4105) / 0001000000001 001	P_5 (4065) / 0000111111100 001	P_6 (4110) / 0001000000001 110
Band3		Band4	
P_1 (4529) / 0001000110110 001	P_2 (5378) / 0001010100000 010	P_1 (4490) / 0001000110001 010	P_2 (5321) / 0001010011001 001
P_3 (5280) / 0001010010100 000	P_4 (4087) / 0000111111110 111	P_3 (4070) / 0000111111100 110	P_4 (3972) / 0000111110000 100
P_5 (4095) / 0000111111111 111	P_6 (4100) / 0001000000000 100	P_5 (4060) / 0000111111011 100	P_6 (4062) / 0000111111011 110

FIGURE 3. Example of recording band changes

After analyzing the HSI data presented in Figure 3, significant redundancy is observed in the higher-order bits, particularly within the first three MSBs of pixels, which are frequently zero. This redundancy indicates that, despite the 16-bit data storage capacity of HSI, the higher-order bits show limited variability across multiple pixels and adjacent bands. This underutilization enables us to efficiently exploit this redundancy as part of our data hiding strategy.

To gain a deeper understanding of the redundancy feature of pixels in HSI, several analyses are conducted. These analyses help to reveal patterns and variations in pixel

values, particularly focusing on higher-order bits and their underutilization across different bands and pixels. First, a statistical analysis of the HSI is conducted, as shown in Figure 4, which tracks the number of consecutive adjacent bands sharing the same value for the fourth MSB of pixels. The numbers in the figure represent the counts of consecutive identical fourth MSB values across bands, with commas indicating breaks. For example, consider pixel P_3 , whose vector is recorded as (1, 2, 1). Specifically, the fourth MSB values of the four consecutive bands for P_3 are 0, 1, 1, and 0, respectively. This sequence indicates the following: (a) The first band of P_3 has a different fourth MSB value compared to the second band, so it is recorded as 1. (b) The second and third bands have identical fourth MSB values, resulting in a count of 2. (c) The fourth band has a different fourth MSB value from the third band, so it is recorded as 1. By applying this procedure to all bands, the vector for P_3 is constructed. Repeating the same process for all pixels allows for determining the number of consecutive bands with identical fourth MSB values across the entire image.

P_1 (1,3)	P_2 (4)
P_3 (1,2,1)	P_4 (2,2)
P_5 (4)	P_6 (3,1)

FIGURE 4. Illustration of MSB contiguous band counts in the current band

As noted earlier, the first three MSBs of most bands in hyperspectral images (HSIs) are consistently zero. Consequently, during the initial statistical process, these MSBs were directly set to zero. To handle rare exceptions, the positions where the first three MSBs are non-zero are recorded separately. These positions are excluded from subsequent data embedding. Specifically, the entire image is scanned to identify pixels where the first three MSBs are not all zero. Each identified position is recorded as a coordinate (i, j, b) , where i and j denote the position, and b indicates the band.

Building on the relationships between adjacent bands, variations between them can be classified using labels. If the fourth MSB of pixels in adjacent bands remains unchanged, it is labeled as “0”; if it differs from the fourth MSB of the previous band, it is labeled as “1”. For example, consider band P_3 , with statistical results illustrated in Figure 5: (a) The second band differs from the first band in the fourth MSB, so the first position is labeled as 1, indicating a change. (b) The second and third bands have identical fourth MSBs, so the second position is labeled as 0, indicating no change. (c) Following this, as shown in Figure 4, the fourth band differs from the previous two consecutive identical bands, so the third position is labeled as 1, indicating a change. (d) Finally, the label “0” is used to mark the end. By applying this process to all bands and pixels, a complete statistical label map is generated, recording the labels for every pixel based on changes in their fourth MSBs of pixels across adjacent bands.

In our approach, we employ an arithmetic encoding technique to compress data and enhance transmission efficiency. The label map, with partial results shown in Figure 5 as an example, is processed row by row. For each row, the unique elements are identified, and their respective frequencies are counted to construct a probability model. The row data is then transformed into an index sequence, where each index represents the position of a value within the set of unique elements. Based on this index sequence and the frequency distribution, arithmetic encoding is applied to generate a compact binary code. As demonstrated in the experimental results presented in Section 4, the average compression

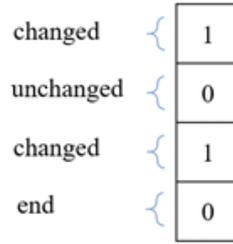


FIGURE 5. Example of recording changes in the fourth MSBs of pixels across adjacent bands

ratio achieved by arithmetic coding on the label map of an HSI is approximately 24.86%, which is calculated by dividing the total length of the encoded code by the total length of the label map before encoding.

3.2. Image Encryption Phase. Before embedding secret data, it is essential to encrypt the HSI to safeguard both the HSI and the secret data from potential leakage. Initially, for the plaintext HSI I , a random matrix R of size $W \times H$ is generated to serve as the encryption key K_e , where the values of K_e are within the range $[0, 2^{16} - 1]$ and pre-shared between the content owner and receiver. Next, the plaintext HSI I and the pseudo-random matrix R are XORed to encrypt the image I , as shown in Eq. (3). Each encrypted 16-bit binary number is then converted to a decimal number according to Eq. (4), producing the encrypted pixel values. This operation is performed for each band in the HSI, resulting in the encrypted HSI I_e . Figure 6 shows the example of hyperspectral image encryption.

$$Ie_{(i,j)}^k = I_{(i,j)}^k \oplus R_{(i,j)}^k \tag{3}$$

$$Ie_{(i,j)} = \sum_{k=1}^{16} Ie_{(i,j)}^k \times 2^{k-1} \tag{4}$$

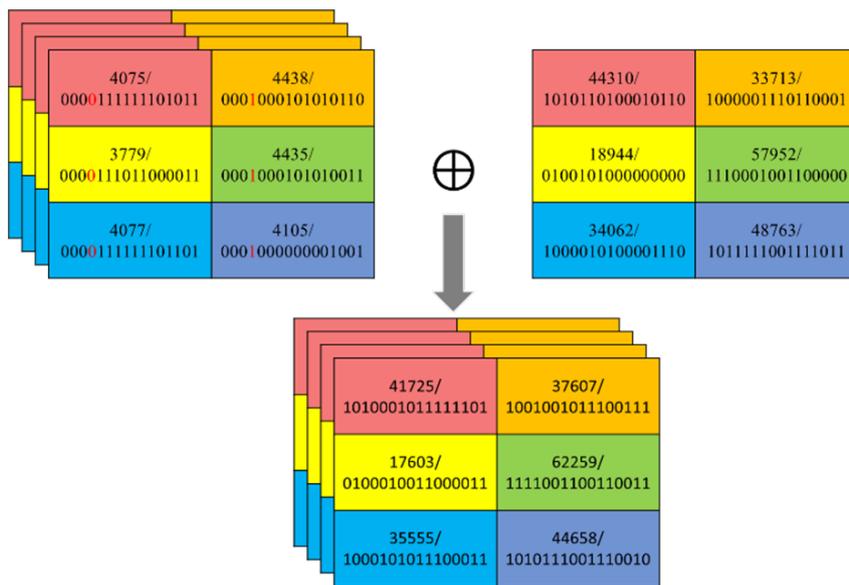


FIGURE 6. Example of hyperspectral image (HSI) encryption

3.3. Auxiliary Data Generation. To ensure the reversibility of the HSI, several types of auxiliary data (AD) must be embedded. As illustrated in Figure 7, the auxiliary data (AD) includes several key components: the encoded code length, the codebook length, the secret data length, the encoded code with the codebook, and any exceptions encountered during the embedding process. Incorporating this auxiliary data enables precise extraction and reconstruction of both the embedded information and the original image. The content and length of the encoded code and codebook are determined during the arithmetic encoding process. The secret message length is derived by directly reading the secret message. Exceptions are identified by scanning the entire hyperspectral image (HSI) to find positions where the first three most significant bits (MSBs) are not all zero, with these positions recorded. This approach ensures the accurate restoration of the image using the corresponding auxiliary information.

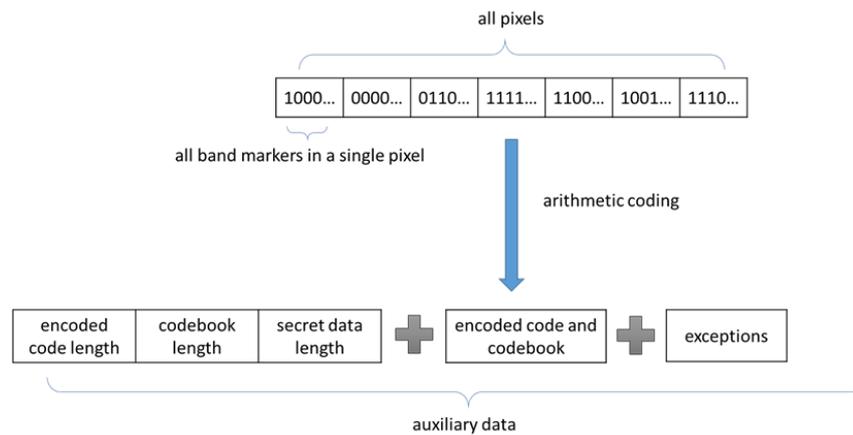


FIGURE 7. The structure of the AD

3.4. Data Encryption and Embedding. Upon receiving the encrypted HSI I_e , the data hider first enhances the security of the embedded information by performing stream encryption on the secret data using the data encryption key K_d . Once the encryption operation is completed, AD is embedded into the encrypted HSI I_e , starting from the second band of the first pixel. After embedding the AD, the encrypted secret information is then embedded.

The HSI is classified into embeddable and non-embeddable regions. The non-embeddable regions consist of two segments: the first segment includes positions where the first three most significant bit (MSB) values are not all zero, as described in Section 3.1. The second segment comprises the first band of each pixel, which serves as a reference band for subsequent bands. The remaining regions are categorized as embeddable, where the MSBs are replaced bit by bit with auxiliary data and secret information. This process continues until all information is successfully embedded into the encrypted image.

As shown in Figure 8, after completing the initial encryption process and incorporating the auxiliary data (AD), the embedding process is performed pixel by pixel, as illustrated in the example. Consider the example in Figure 3: the first band is used as the reference, while the other three bands are processed. For the three bands at position (1,1), the first band remains unchanged, and the four most significant bits (MSBs) of the other bands are directly replaced with the content, which includes the AD and secret data to be embedded. In Figure 8, different colors are used to differentiate between pixels. For example, red represents bands 1 to 4 at position (1,1), while orange represents bands 1 to 4 at position (1,2). Once the embedding is completed at position (1,1), the process

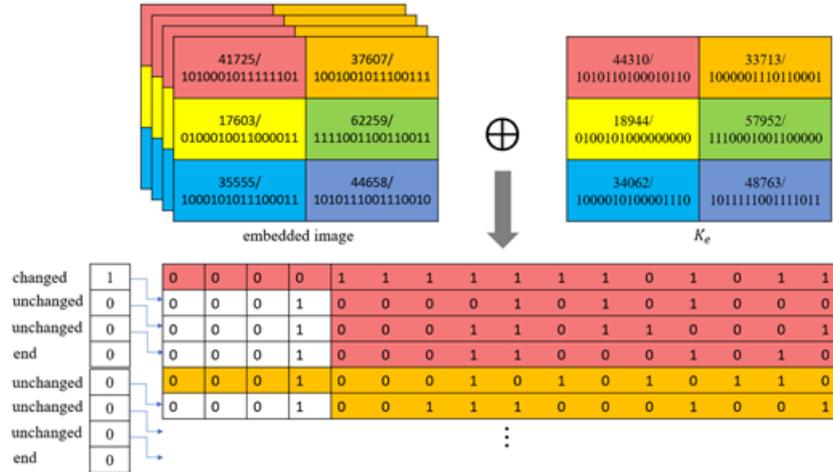


FIGURE 9. The example of image recovery

4. Experimental results and analyses. This section evaluates the effectiveness of the proposed scheme. To demonstrate the performance of our scheme alongside that of X. Zhang et al. [26], we used six hyperspectral test images: “Salinas,” “Pavia University,” “Indian Pines,” “Botswana,” “Cuprite,” and “KSC,” as shown in Figure 10. The dimensions of these images are as follows: the first image measures $512 \times 217 \times 204$, the second image $1096 \times 715 \times 102$, the third image $145 \times 145 \times 200$, the fourth image $1476 \times 256 \times 145$, the fifth image $512 \times 614 \times 224$, and the sixth image $512 \times 614 \times 176$. All computations were conducted on a PC equipped with a 3.6GHz Intel(R) Core (TM) i7-4790 CPU and 8 GB of RAM, running Windows 10 Professional. The algorithms were implemented in MATLAB R2023a.

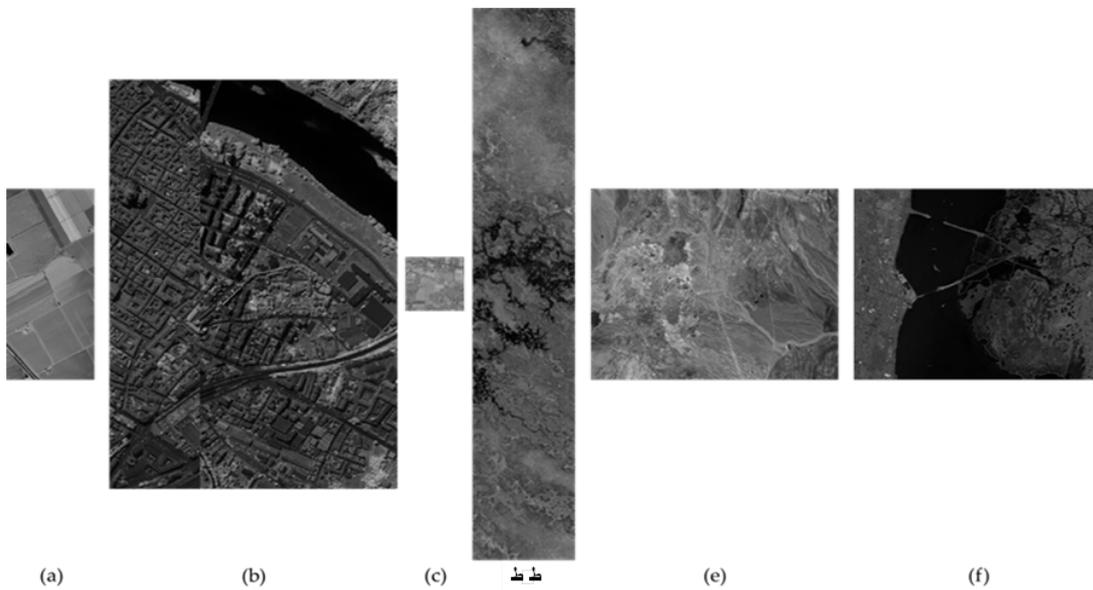


FIGURE 10. One selected band of three hyperspectral test images. a) Salinas; b) Pavia University; c) Indian pines; d) Botswana; e) Cuprite; f) KSC

To validate the feasibility of our proposed design, we analyzed the similarity among the first six most significant bits (MSBs) in neighboring bands across three datasets. As shown in Table 2, we evaluated the similarity of three hyperspectral images using different

MSB lengths. In the table, the numbers 1 to 6 indicate whether the first 1 to 6 MSBs of adjacent bands are identical. The data represents the number of instances where no changes occur between adjacent bands for a single pixel.

For example, if a pixel consists of 100 bands, where the first 50 bands have the same value, and the next 50 bands also share the same value, with a change occurring only between the 50th and 51st bands, the unchanged band count for this pixel would be 99. By summing the unchanged counts of all pixels, we calculated the proportion of unchanged bands relative to the total number of bands. This proportion, representing the similarity, is presented in Table 2. The results confirm that the similarity among the first four MSBs of neighboring bands is remarkably high.

TABLE 2. The similarity of each data set under different digits of MSB

Number of MSB bits	Salinas	Pavia University	Indian Pines	Botswana	Cuprite	KSC
1	100%	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%	100%
3	99.99%	100%	99.95%	100%	99.54%	100%
4	99.21%	99.87%	93.34%	95.86%	99.54%	100%
5	93.55%	98.67%	87.39%	89.66%	94.95%	100%
6	85.64%	95.87%	72.71%	77.93%	90.83%	100%

To further verify the usability of the proposed data hiding strategy, we assess the necessity of evaluating the similarity between adjacent bands at the bit level. Two adjacent frequency bands are randomly selected as a pair and the prior band is used as a reference to compute the PSNR of the band pair. In Table 3, five band pairs are selected from three test HIS images, and their PSNR values are calculated. The results indicate that although the pixel values of adjacent bands are often highly similar which are ranging between [37.74 dB and 51.46 dB], rare cases where the PSNR falls below 25 dB still exist. This finding validates the necessity of performing data hiding on adjacent bands at the bit level using our proposed scheme.

TABLE 3. Comparison of PSNR (db) between randomly selected adjacent spectral bands in hyperspectral images

	Pair1	Pair2	Pair3	Pair4	Pair5
Salinas	43.10	38.75	46.55	34.33	39.98
Pavia University	45.24	50.18	51.46	46.71	48.61
Indian Pines	24.31	37.74	35.91	26.11	45.93

To better illustrate the advantages of our scheme, we evaluated its effectiveness from two perspectives: whether the MSB label is used and whether arithmetic coding is applied. The results, shown in Table 4, indicate that when arithmetic coding is not used, the effectiveness remains the same regardless of whether the MSB label is adopted. This is because we use symbols of the same length to represent MSB values, resulting in the same amount of freed space. However, when arithmetic coding is applied, the use of the MSB label becomes crucial.

This is mainly due to the correlation between spectral bands. For example, in an HSI with 100 bands, if the first 50 bands have identical values and the last 50 bands also share the same values, with changes occurring only between the 50th and 51st bands, then 99% of the MSB label remains identical. This allows for more efficient compression

during encoding. Conversely, without the MSB label, up to 50% of the band values differ, leading to less effective encoding compared to using the MSB label.

TABLE 4. Ablation study of embedding capacity (bits) on the six HSIs

Measurements	Images	No MSB Label	No MSB Label	MSB Label	MSB Label
		No Arithmetic	Arithmetic	No Arithmetic	Arithmetic
Total capacity	Salinas	6.80×10^7	9.31×10^7	6.80×10^7	1.06
	Pavia University	2.40×10^8	3.12×10^8	2.40×10^8	3.80
	Indian Pines	1.26×10^7	1.42×10^7	1.26×10^7	1.88
	Botswana	1.64×10^8	2.04×10^8	1.64×10^8	2.46
	Cuprite	2.11×10^8	2.84×10^8	2.11×10^8	3.33
	KSC	1.66×10^8	2.70×10^8	1.66×10^8	2.71
Embedding rate (bpppb)	Salinas	3	4.11	3	4.0
	Pavia University	3	3.90	3	4.7
	Indian Pines	3	3.37	3	4.4
	Botswana	3	3.73	3	4.4
	Cuprite	3	4.03	3	4.7
	KSC	3	4.88	3	4.4

To demonstrate the advantages of our proposed scheme by leveraging the similarity of MSBs among neighboring bands, we compared it with Zhang et al.’s approach [26]. As our technique is the first to perform data hiding in hyperspectral images (HSI) within the encrypted domain, we created a simplified version of our scheme, referred to as “Direct,” by omitting the label map mechanism. The results of these comparisons are presented in Table 5.

In Table 5, “Direct” represents the use of arithmetic coding to directly encode the MSB values of the HSI without employing the label map mechanism. The “4MSB” approach evaluates the first four MSBs of the bands, encoding values such as 0000 as 0, 0001 as 1, and so forth. Similarly, “5MSB” encodes values from 00000 to 00011 as 0 to 3 using arithmetic coding. As shown, our proposed scheme consistently achieves higher embedding capacity across all HSIs. For instance, the average embedding rate of our scheme is 3.76 bits per pixel per band (bpppb), significantly outperforming Zhang et al.’s scheme, which achieves only 0.36 bpppb. This comparison further validates the effectiveness and superiority of our approach. The bpppb metric is calculated by dividing the total embedding capacity by the product of the dimensions W , H , and B of the HSI, resulting in the embedding capacity per pixel in a single band.

In Table 6, we compare the maximum embedding capacity reported by Zhang et al. [26] after removing the total number of bands with the Embedding rate (bpppb). Considering that Zhang et al.’s method [26] has three embedding modes, each with different embedding capacities depending on the block size, Table 6 further compares the maximum embedding capacity reported in Zhang et al.’s paper for each embedding mode with our proposed scheme. From Table 4, we can see that our proposed scheme significantly outperforms [26] in terms of embedding capacity. This is because our method is based on reversible data hiding in the encrypted domain (RDHEI). Unlike [26], which is limited by image quality in the plaintext domain and the limitations inherent in their use of AMBTC, we can fully leverage the characteristics of the encrypted domain to create more space for embedding. Additionally, we first explore the bit value similarity between adjacent bands’ pixels, and

TABLE 5. Embedding capacity (bits) of each method

Measurements	Methods	Salinas	Pavia University	Indian Pines	Botswana
Total capacity	Proposed (4MSB)	8.76×10^7	3.04×10^8	1.51×10^7	2.02×10^8
	Proposed (5MSB)	1.06×10^8	3.80×10^8	1.88×10^7	2.46×10^8
	Direct (4MSB)	8.61×10^7	2.64×10^8	1.34×10^7	1.95×10^8
	Direct (5MSB)	9.31×10^7	3.12×10^8	1.42×10^7	2.04×10^8
	Zhang et al. [26]	4.19×10^6	1.89×10^7	2.78×10^6	-
Embedding rate (bpppb)	Proposed (4MSB)	3.86	3.8	3.59	3.69
	Proposed (5MSB)	4.68	4.75	4.47	4.49
	Direct (4MSB)	3.80	3.30	3.19	3.57
	Direct (5MSB)	4.11	3.90	3.38	3.73
	Zhang et al. [26]	0.18	0.24	0.66	-

Note: “-” indicates no data has been reported in Zhang et al. [26].

based on this similarity, we designed our method, which is why our approach has made this breakthrough in embedding capacity.

TABLE 6. Maximum embedding capacity (bits) of each scheme

Methods	Salinas		Pavia University		Indian Pines	
	2×2	4×4	2×2	4×4	2×2	4×4
Zhang et al. [26]	Type1	3.07×10^6	2.90×10^6	1.29×10^7	1.41×10^7	1.98×10^7
	Type2	3.58×10^6	2.22×10^6	1.58×10^7	1.14×10^7	2.32×10^7
	Type3	4.19×10^6	3.00×10^6	1.78×10^7	1.89×10^7	2.78×10^7
Ours	Direct (5MSB)	9.31×10^7	3.12×10^8	1.42×10^7	-	-
	Proposed (5MSB)	1.06×10^8	3.80×10^8	1.88×10^7	-	-

To evaluate the effectiveness of the encryption process, the number of pixels change rate (NPCR), unified average changing intensity (UACI), and peak signal-to-noise ratio (PSNR) analyses were conducted on the images before and after encryption, as summarized in Table 7. NPCR is calculated using the formula in Eq. (5).

$$\text{NPCR} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H D(i, j) \times 100\% \quad (5)$$

where $D(i, j) = 1$ if the pixel values at position (i, j) are different in the original and encrypted images, otherwise $D(i, j) = 0$. A higher NPCR indicates that the encryption method effectively alters the pixel values, enhancing the security of the encrypted image. The ideal NPCR value is greater than 99%. UACI is calculated as in Eq. (6).

$$\text{UACI} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H \frac{|C(i, j) - P(i, j)|}{65535} \times 100\% \quad (6)$$

where $P(i, j)$ and $C(i, j)$ denote the pixel values of the original and encrypted images, respectively. A higher UACI value implies greater intensity differences, contributing to enhanced resistance against attacks. The ideal UACI value is around 33%. From Table 7, I can observe that after encryption, the NPCR of the six test images is greater than 99%, and the UACI is around 48%. Additionally, the PSNR after encryption is only about 5

dB. Therefore, we can conclude that the block-based encryption method employed in our approach provides a certain level of security, while the original pixel information is not preserved visually.

TABLE 7. NPCR, UACI, and PSNR Analysis of Images Before and After Encryption

	NPCR (%)		UACI (%)		PSNR (dB)		
	MAX	MIN	Mean	MAX	MIN	Mean	
Salinas	99.9991	99.9991	99.9991	49.9940	45.0428	48.2532	5.00
Pavia University	99.9986	99.9986	99.9986	49.0636	47.5196	48.3772	4.99
Indian Pines	99.9952	99.9952	99.9952	48.3062	41.2542	45.9932	5.33
Botswana	99.9984	99.9984	99.9984	49.8759	43.4545	47.4699	5.11
Cuprite	99.9997	99.9981	99.9981	50.0596	42.2512	46.3828	5.27
KSC	99.9990	99.9990	99.9990	50.0107	49.7630	49.8925	4.78

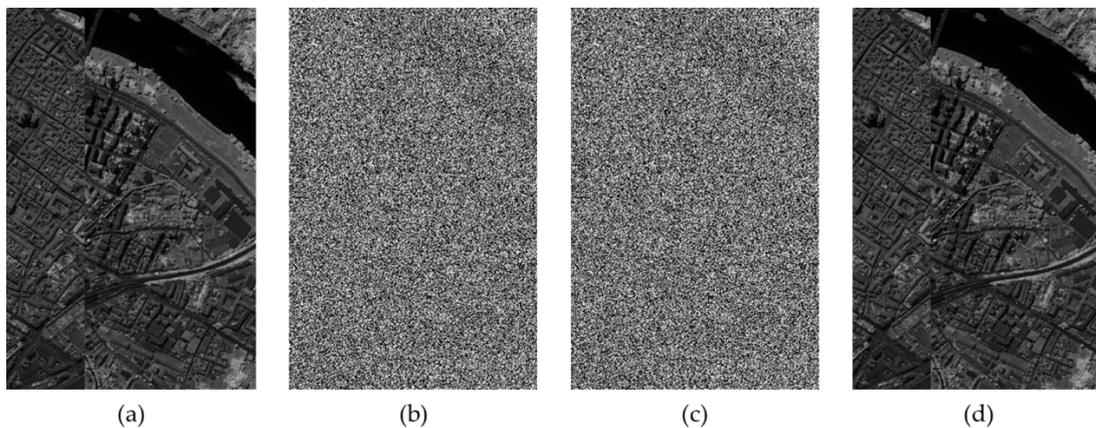


FIGURE 11. Results of our method executed on Pavia. a) Original image; b) Encrypted image; c) Encrypted and embedded image; d) Restored image

Unlike Table 7, Figure 12 presents the line charts of NPCR and UACI for the encrypted images. Notably, in the Cuprite dataset, four completely black bands appear in bands 108 to 111, and twelve completely black bands appear in bands 155 to 166. This explains why the UACI values in the Cuprite line chart exhibit one short and one long significant fluctuation. This phenomenon also highlights the necessity of the bit-level similarity assessment adopted in our method, as discussed in Tables 2 and 3, as well as the usability of our bit-level-based data hiding approach.

Regarding space complexity, the primary data structures include the original image, its preprocessed binary representation, and various intermediate results (e.g., MSB statistics and encoding outputs), all of which are proportional to the image size $M \times N \times B$. As a result, the space complexity of our method is $O(M \times N \times B)$. From the analysis above, it is clear that as the image size or number of bands increases, the computation time is mainly affected by the sorting operations in the arithmetic encoding and decoding process, while the memory requirement grows linearly with the size of the image data. Note that, except for the schemes by Zhang et al. [26] and ours, which are designed for HSI, the schemes by Huang et al. [27], Ji et al. [28], and Yao et al. [29] are either dual image-based RDH [27] or RDHEI schemes [28, 29]. The hiding capacity for these three schemes is measured

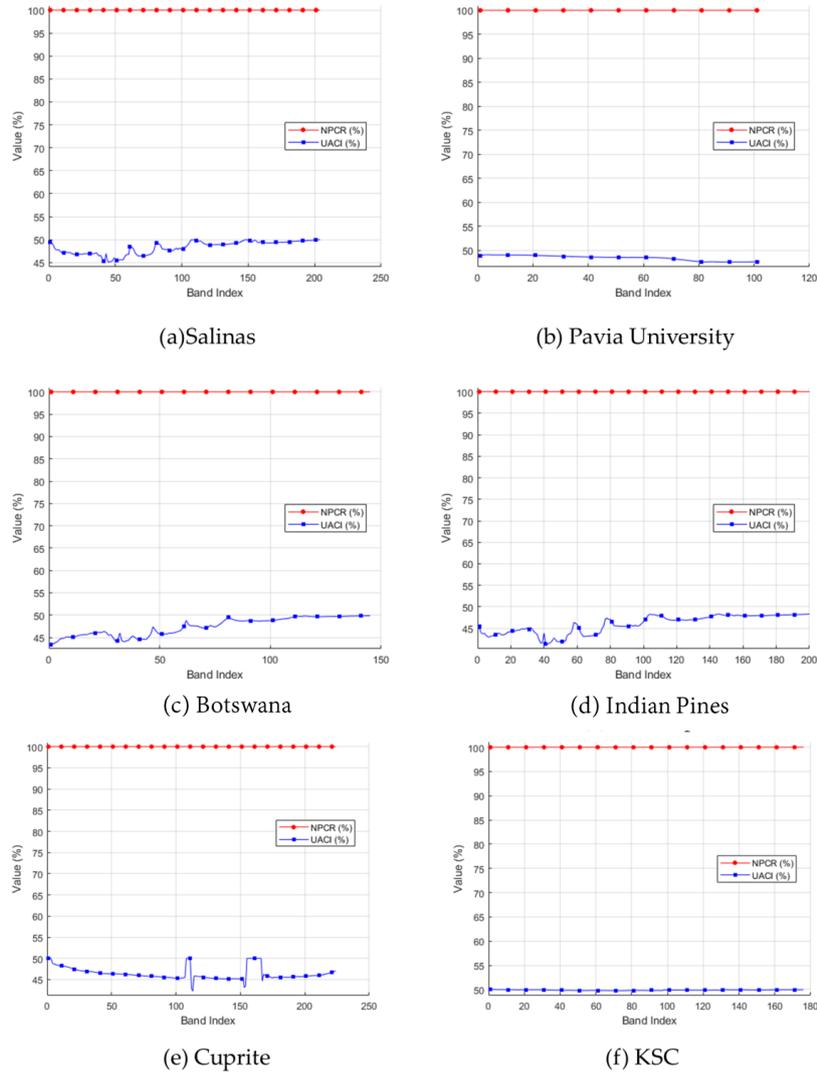


FIGURE 12. Line Charts of NPCR and UACI for Different Images

in bpp (bits per pixel), whereas, for Zhang et al. [26] and our scheme, it is measured in bpppb (bits per pixel per band). Among the five schemes, the highest hiding capacity reaches 4.9 bpppb while maintaining the same space complexity as Zhang et al. [26], with only a slight increase in time complexity.

TABLE 8. Analysis of time, space complexity and hiding capacity (bpp/bpppb)

Methods	Time complexity	Space complexity	Hiding capacity
Zhang et al. [26]	$O(M \times N \times B)$	$O(M \times N \times B)$	4.25 (Selected images)
Huang et al. [27]	$O((M \times N \times B)^2)$	$O(M \times N \times B)$	3.9387 (BOSSBase)
Ji et al. [28]	$O((M \times N \times B)^2)$	$O((M \times N \times B)^2)$	3.9387 (BOSSBase)
Yao et al. [29]	$O((M \times N \times B)^2)$	$O((M \times N \times B)^2)$	3.793 (BOSSBase)
Proposed	$O(M \times N \times B \log B)$	$O(M \times N \times B)$	[3.90-4.9] (Selected images)

5. Conclusion. In this research, we proposed a novel data hiding method for hyperspectral images (HSI), termed RDHEHI-MMSB. This method leverages the inherent redundancy between spectral bands to design an innovative encoding and embedding mechanism. Our scheme begins with an analysis of the redundancy in the most significant bits (MSBs) of the HSI, enabling efficient embedding of secret information. Experimental results demonstrate that our data hiding strategy, which leverages bit-level similarity among adjacent bands, achieves an embedding capacity of up to 3.76 bits per pixel per band (bpppb), significantly outperforming existing methods. Additionally, the block-based key encryption ensures the robustness of the encrypted HSI, preventing any information leakage to potential attackers. Comparative evaluations further reveal that our method excels in embedding capacity, image reconstruction quality, and secret data extraction accuracy.

Key advantages of our approach include its ability to securely hide data while maintaining high fidelity in image restoration, making it ideal for applications where both security and image quality are paramount. Future research could explore enhancing the algorithm's efficiency and robustness against potential attacks using either deep learning approaches [30] or conventional techniques. Additionally, integrating machine learning techniques presents promising opportunities to further optimize data hiding processes within hyperspectral imaging frameworks. Establishing or obtaining access to a hyperspectral image database will also be crucial for our future studies. Moreover, our experiments have identified rare cases where adjacent bands are not similar to each other, posing another challenge. If such a phenomenon becomes more prevalent, developing an indexing strategy to systematically link similar bands will become essential, making it a key focus of our future work.

REFERENCES

- [1] Chang, C.C., Nguyen, T.S. and Lin, C.C., 2015. A reversible compression code hiding using SOC and SMVQ indices. *Information Sciences*, 300, pp.85-99.
- [2] Ma, X., Pan, Z., Hu, S. and Wang, L., 2015. Reversible data hiding scheme for VQ indices based on modified locally adaptive coding and double-layer embedding strategy. *Journal of Visual Communication and Image Representation*, 28, pp.60-70.
- [3] Zhang, T., Weng, S., Wu, Z., Lin, J. and Hong, W., 2022. Adaptive encoding based lossless data hiding method for VQ compressed images using tabu search. *Information Sciences*, 602, pp.128-142.
- [4] Wang, L., Pan, Z., Ma, X. and Hu, S., 2014. A novel high-performance reversible data hiding scheme using SMVQ and improved locally adaptive coding method. *Journal of Visual Communication and Image Representation*, 25(2), pp.454-465.
- [5] Pan, Z., Ma, X., Deng, X. and Hu, S., 2013. Low bit-rate information hiding method based on search-order-coding technique. *Journal of Systems and Software*, 86(11), pp.2863-2869.
- [6] Tian, J., 2003. Reversible data embedding using a difference expansion. *IEEE transactions on circuits and systems for video technology*, 13(8), pp.890-896.
- [7] Ni, Z., Shi, Y.Q., Ansari, N. and Su, W., 2006. Reversible data hiding. *IEEE Transactions on circuits and systems for video technology*, 16(3), pp.354-362.
- [8] Pan, Z., Hu, S., Ma, X. and Wang, L., 2015. Reversible data hiding based on local histogram shifting with multilayer embedding. *Journal of Visual Communication and Image Representation*, 31, pp.64-74.
- [9] Ma, X., Pan, Z., Hu, S. and Wang, L., 2015. High-fidelity reversible data hiding scheme based on multi-predictor sorting and selecting mechanism. *Journal of Visual Communication and Image Representation*, 28, pp.71-82.
- [10] Zhang, T., Hou, T., Weng, S., Zou, F., Zhang, H. and Chang, C.C., 2022. Adaptive reversible data hiding with contrast enhancement based on multi-histogram modification. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8), pp.5041-5054.
- [11] Weng, S., Tan, W., Ou, B. and Pan, J.S., 2021. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. *Information Sciences*, 549, pp.13-33.

- [12] Fan, G., Pan, Z., Gao, E., Gao, X. and Zhang, X., 2021. Reversible data hiding method based on combining IPVO with bias-added Rhombus predictor by multi-predictor mechanism. *Signal Processing*, 180, p.107888.
- [13] Fan, G., Pan, Z., Zhou, Q., Gao, X. and Zhang, X., 2021. Multiple histogram based adaptive pairwise prediction-error modification for efficient reversible image watermarking. *Information Sciences*, 581, pp.515-535.
- [14] Fan, G., Pan, Z., Zhou, Q. and Zhang, X., 2023. Flexible patch moving modes for pixel-value-ordering based reversible data hiding methods. *Expert Systems with Applications*, 214, p.119154.
- [15] Pan, Z., Gao, X., Gao, E. and Fan, G., 2020. Adaptive complexity for pixel-value-ordering based reversible data hiding. *IEEE Signal Processing Letters*, 27, pp.915-919.
- [16] Weng, S., Zhang, C., Zhang, T. and Chen, K., 2021. High capacity reversible data hiding in encrypted images using SIBRW and GCC. *Journal of Visual Communication and Image Representation*, 75, p.102932.
- [17] Huang, F., Huang, J. and Shi, Y.Q., 2016. New framework for reversible data hiding in encrypted domain. *IEEE Transactions on Information Forensics and Security*, 11(12), pp.2777-2789.
- [18] Rad, R.M., Wong, K. and Guo, J.M., 2014. A unified data embedding and scrambling method. *IEEE Transactions on Image processing*, 23(4), pp.1463-1475.
- [19] Chang, C.C., Nguyen, T.S. and Lin, C.C., 2013. Distortion-free data embedding scheme for high dynamic range images. *Journal of Electronic Science and Technology*, 11(1), pp.20-26.
- [20] Landgrebe, D., 2002. Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19(1), pp.17-28.
- [21] Shetty, N.R., 2024, July. A study and analysis of reversible data hiding techniques. In *2024 Second International Conference on Advances in Information Technology (ICAIT)* (Vol. 1, pp. 1-6).
- [22] Wang, J. and Ou, B., 2024. Video reversible data hiding: A systematic review. *Journal of Visual Communication and Image Representation*, 98, p.104029.
- [23] Zhang, X., 2011. Reversible data hiding in encrypted image. *IEEE signal processing letters*, 18(4), pp.255-258.
- [24] Zhang, M., Tong, X.J., Liu, J., Wang, Z., Liu, J., Liu, B. and Ma, J., 2020. Image compression and encryption scheme based on compressive sensing and Fourier transform. *IEEE Access*, 8, pp.40838-40849.
- [25] Singh, R. and Vaish, A., 2021. MSB/LSB prediction based reversible data hiding in encrypted images: a survey. *Machine Intelligence and Smart Systems: Proceedings of MISS 2020*, pp.11-24.
- [26] Zhang, X., Pan, Z., Zhou, Q., Fan, G. and Dong, J., 2024. A reversible data hiding method based on bitmap prediction for AMBTC compressed hyperspectral images. *Journal of Information Security and Applications*, 81, p.103697.
- [27] Huang, C.T., Weng, C.Y. and Shongwe, N.S., 2023. Capacity-raising reversible data hiding using empirical plus-minus one in dual images. *Mathematics*, 11(8), p.1764.
- [28] Ji, C., Gao, G. and Shi, Y.Q., 2023. Reversible data hiding in encrypted images with adaptive Huffman code based on dynamic prediction axes. *IEEE Transactions on Multimedia*, 26, pp.5962-5975.
- [29] Yao, Y., Wang, K., Chang, Q. and Weng, S., 2023. Reversible data hiding in encrypted images using global compression of zero-valued high bit-planes and block rearrangement. *IEEE Transactions on Multimedia*, 26, pp.3701-3714.
- [30] Hyun Kwon, 2024. AudioGuard: Speech recognition system robust against optimized audio adversarial examples. *Multimedia Tools and Applications*, 83, pp. 57943-57962.